Doctoral theses at NTNU, 2022:126

Evelyn Honoré-Livermore

Integrating Agile Systems Engineering and Project Management in Small Satellites Development

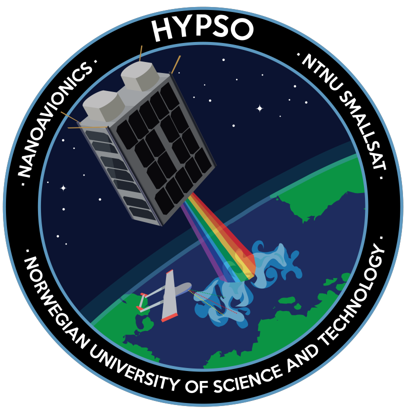
Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of Philosophiae Doctor
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of Science and Technology

Evelyn Honoré-Livermore

Integrating Agile Systems Engineering and Project Management in Small Satellites Development



Thesis for the Degree of Philosophiae Doctor

Trondheim, May 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

ISBN 978-82-326-6888-5 (printed ver.)
ISBN 978-82-326-6180-0 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2022:126

Printed by NTNU Grafisk senter

Abstract

This dissertation aims to improve our understanding of applied research projects in academia, in particular, how the integration of Systems Engineering (SE) and Project Management (PM) activities can support these projects. This is studied in the context of a university CubeSat organization called HYPSO, which aims at contributing scientific data and operational capabilities in a network of autonomous systems to improve our monitoring and understanding of the oceans. This network is a part of the Mission-oriented Autonomous Systems with Small Satellites for Maritime Sensing, Surveillance and Communication (MASSIVE) project. The overarching goal of the project organization is to reduce the time between the mission concept is defined and launch while still ensuring project success, through changing and adapting their development methodology and processes. The project is highly interdisciplinary and produces complex engineered systems. SE and PM offer heuristics for addressing complexity, but require resources and competencies not found in many academic research projects. Little empirical research was found on how universities conduct applied research projects, and to what degree SE and PM practices are applied. The research aim and goal gave the following research questions:

RQ-0: What are known sociotechnical challenges in university CubeSat projects and how can they be addressed?
RQ-1: What factors influence the development time of university CubeSat systems?
RQ-2: To what extent do agile processes support known university CubeSat project challenges linked to knowledge management, system testing, project management, and team composition?
RQ-3: To what extent can Model-Based Systems Engineering support university CubeSat projects and the development of System-of-Systems?

These are addressed based on the findings from (i) a longitudinal replication case study of a university CubeSat team consisting of long-term PhD-level researchers and students that join the project for the duration of a schoolyear; and (ii) 18 semi-structured interviews. Participatory Action Research and interviews are the primary methods for collecting the data. The analysis is supported by system modeling methods and tools such as Model-Based Systems Engineering (MBSE), Systemigrams, Agile Decision-Guidance, Readiness Levels, and N2-diagrams. The research questions are addressed through the ten publications included in this thesis.

RQ-0 is addressed in the literature review, and supported by findings from the case study. The literature review highlights sociotechnical challenges encountered when conducting CubeSat projects at universities: project management, team and organization, balancing schoolwork with project work, ensuring mission success, knowledge management, and stakeholder management.

RQ-1 goes deeper into the case study, developing an understanding of what factors affect the development time of space systems in university CubeSat projects. A combination of analysis methods highlights the following factors: **external facilities and support environment**, **communication and information flow**, **goal alignment and clear objectives**, **work planning** and **knowledge management**.

In **RQ-2**, this dissertation looks at how agile SE and PM practices meet these challenges, by using the HYPSO project as a replication case study. Agile values (often operationalized by Scrum or eXtreme Programming frameworks) promote “Individuals and interactions over processes and tools”, “Working software over comprehensive documentation”, “Customer collaboration over contract negotiation”, and “Responding to change over following a plan” [1].

Findings from the case study show that the project exhibits many characteristics suitable for an agile approach, as opposed to a directed plan approach. A tailored Scrum approach enables improved planning, communication, alignment of goals, and responsiveness to stakeholders, changing working situations, external facilities and the support environment. In addition, the project was well supported by digital tools and workflows, contributing to moving the organization towards adopting Digital Engineering.

RQ-3 investigates to what extent MBSE practices can support university CubeSat projects and address some of the challenges identified in **RQ-0** and **RQ-1**. MBSE is hailed as a paradigm shift in SE, improving the workflow by enabling system designers and stakeholders to describe the complex system-of-interest from different viewpoints, without losing the semantic relationship between the operational, logical, physical, and system viewpoints.

Findings from the application of MBSE in the case study show possible improvements in traceability of dependability analysis and system design, when compared to document-based SE. There is also potential to reuse dependability and system design for future satellites, which could lower the resource needs moving forward. The findings reported in this thesis indicate that taking a Systems-of-Systems (SoS) approach to viewing the MASSIVE project supported the design process of the different Constituent Systems (CS) and identified missing functionality and interfaces by establishing an overall SoS architecture and concept of operations.

This research contributes to both theory and practice by showing how an integrated approach to agile Systems Engineering and Project Management supported by Digital Engineering tools and methods can improve the way applied research projects are managed at universities, specifically when developing systems for space. In addition, the research contributes with experiential evidence of the value of using an SoS viewpoint to coordinate the development efforts and integration of multiple assets. This dissertation supports the efforts of adopting MBSE, and further towards Digital Engineering.

Sammendrag

Denne avhandlingen tar sikte på å forbedre vår forståelse av anvendte forskningsprosjekter i akademia, og fokuserer på hvordan integrering av aktiviteter kan støtte og forbedre disse prosjektene. En prosjektorganisasjon ved NTNU, kalt HYPSO, blir studert i sammenheng med dette. Organisasjonen tar sikte på å bidra med vitenskapelige data og operasjonelle kapasiteter i et nettverk av autonome systemer, for å forbedre vår overvåking og forståelse av havene. Dette nettverket er en del av prosjektet MASSIVE. Det overordnede målet for prosjektorganisasjonen er å redusere tiden mellom konseptdefinisjon og oppskyting av satellitter, samt å sikre suksess for prosjektet gjennom å endre og tilpasse utviklingsmetodikk og prosesser. Prosjektet er svært tverrfaglig og produserer komplekse systemer. "Systems Engineering" (systemteknikk (SE)) og "Project Management" (prosjektledelse (PM)) tilbyr heuristikk for å håndtere kompleksitet, men dette krever ressurser og kompetanse som vanligvis ikke finnes i akademiske forskningsprosjekter. Lite empirisk forskning ble funnet om hvordan universiteter gjennomfører anvendte forskningsprosjekter, og i hvilken grad SE og PM-praksis brukes. Forskningsmålene ga følgende forskningsspørsmål:

RQ-0: Hva er kjente sosiotekniske utfordringer for CubeSat-prosjekter ved universiter, og hvordan kan de løses?
RQ-1: Hvilke faktorer har innvirkning på utviklingstid av CubeSat-systemer på universiteter?
RQ-2: I hvilken grad kan agile (smidige) prosesser brukes til å adressere kjente sosiotekniske utfordringer knyttet til kunnskapshåndtering, systemtesting, prosjektledelse, og gruppesammensetning?
RQ-3: I hvilken grad kan Model-Based Systems Engineering (MBSE) støtte CubeSat-prosjekter og utvikling av system-av-systemer på universiteter?

Disse forskningsspørsmålene behandles ved hjelp av funnene fra (i) en langsgående casestudie av et universitets-CubeSat-team bestående av forskere på doktorgradsnivå, samt studenter som er med i prosjektet gjennom et skoleår; og (ii) 18 semistrukturerte intervjuer. Deltakerbasert forskning og intervjuer er hovedmetodene for innsamling av data. Analysen støttes av systemmodelleringsmetoder og verktøy som "Model-Based Systems Engineering (MBSE)", "Systemigrams", "Agile Decision-Guidance", "Readiness Levels" og "N2-diagrammer." Forskningsspørsmålene behandles gjennom de ti publikasjonene som er inkludert i denne oppgaven.

RQ-0 er diskutert i litteraturgjennomgangen, og støttet av funn fra casestudien. Litteraturgjennomgangen belyser sosiotekniske utfordringer man støter på når man gjennomfører CubeSatprojekter ved universiteter: prosjektledelse, team og organisasjon, balansering av skolearbeid med prosjektarbeid, sikring av missionsuksess, kunnskapshåndtering og interessentstyring.

RQ-1 går dypere inn i casestudien, og bidrar til å utvikle en forståelse av hvilke faktorer som påvirker utviklingstiden for romsystemer i CubeSat-prosjekter ved universiteter. Analysen fremhever følgende faktorer: Eksterne fasiliteter og støttemiljø, kommunikasjon og informasjonsflyt, måljustering og klare mål, arbeidsplanlegging og kunnskapsstyring.

RQ-2, ser på hvordan smidig SE og PM praksis møter disse utfordringene. Agile (ofte operasjonalisert av Scrum- eller eXtreme Programming -rammer) prioriterer "individer og interaksjoner fremfor prosesser og verktøy", "fungerende programvare fremfor omfattende dokumentasjon", "kundesamarbeid fremfor kontraktsforhandlinger" og "respons på endring fremfor å følge en plan" [1]. Funn fra casestudien viser at prosjektet har mange egenskaper som er egnet til en smidig tilnærming, i motsetning til en tradisjonell fasestyrt tilnærming. Gjennom en skreddersydd Scrum-tilnærming, muliggjør en forbedret planlegging, kommunikasjon, tilpasning av mål og lydhørhet overfor interessenter, endrede arbeidssituasjoner, eksterne fasiliteter og støttemiljø. Videre støttes den smidige tilnærmingen godt av digitale verktøy og arbeidsflyter, noe som bidrar til å modne organisasjonen for å ta i bruk Digital Engineering.

RQ-3 undersøker i hvilken grad MBSE-praksis kan støtte CubeSat-prosjekter ved universiteter og løse noen av utfordringene identifisert i RQ-0 og RQ-1. MBSE blir hyllet som et paradigmeskifte i SE, som nå beveger seg bort fra dokumentbasert SE og heller forbedrer arbeidsflyten ved å gjøre systemdesignere og interessenter i stand til å beskrive systemet fra forskjellige synspunkter, uten å miste det semantiske forholdet mellom operative, logiske, fysiske og systematiske synspunkter. Funn fra bruken av MBSE i casestudien viser mulige forbedringer i sporbarheten av pålitelighets-analyse og systemdesign. Det er også potensial for å gjenbruke pålitelighet og systemdesign for fremtidige satellitter, noe som kan redusere ressursbehovet fremover. Funnene som er rapportert i denne oppgaven indikerer at det å bruke en tilnærming for system-av-systemer (SoS) for MASSIVE-prosjektet er positivt. Det støttet designprosessen for de forskjellige "Constituent Systems (CS)" og identifiserte manglende funksjonalitet og definisjon av grensesnitt ved å etablere en overordnet SoS-arkitektur og operasjonelt konsept.

Den utførte/presenterte forskningen bidrar til både teori og praksis: Den viser hvordan en integrert tilnærming til smidig systemteknikk og prosjektledelse, støttet av digitale ingeniørverktøy og -metoder, kan forbedre måten anvendte forskningsprosjekter styres ved universiteter, spesielt når man utvikler romfartssystemer. I tillegg bidrar forskningen med empirisk bevis på verdien av å bruke et SoS-synspunkt for å koordinere utviklingsarbeidet og integreringen av flere systemer. Arbeidet presentert i denne avhandlingen støtter arbeidet med å bevege oss bort fra dokumentbasert systems engineering til modellbasert systems engineering, og videre mot en integrert Digital Engineering arbeidsmetodikk.

Preface

My background in project management and systems engineering of space systems in the Norwegian industry, coupled with a passion for understanding how people interact and work together in technical teams, fueled my drive to pursue this PhD. The research direction for this PhD has changed since it started in 2017. When I applied for the research project, it was called “Rapid Systems Engineering for Small Satellites” and was associated with the Mission-oriented Autonomous Systems with Small Satellites for Maritime Sensing, Surveillance and Communication (MASSIVE) project.

During the first months of the research it became clear that what was needed, was to understand how the organization could concurrently execute applied research projects alongside engineering projects and achieve a shorter timeframe from ideation to flight-ready systems for small satellites. Also, there were not many Systems Engineering (SE) activities recognized at the beginning which made it challenging to research *rapid* SE. There seemed to be a confusion between SE and Project Management (PM) understanding and how they were inter-related in delivering projects and products on time. Much of the research design was opportunity-based since the PhD was established to work with the MASSIVE project. This opportunity resulted in the HYPSO project as a replication case study, learning from and comparing with other university CubeSat projects. I built on my own industrial experience as project manager and systems engineer in the traditional space business for the action research conducted in the thesis when acting as project manager for the HYPSO project. The methodology was chosen based on the insights gained through a PhD course called “PK8210 — Systems Engineering Principles and Practice” and developed throughout the PhD research project.

However, even though the title no longer says, “Rapid Systems Engineering for Small Satellites,” the objectives remain the same: to understand how our organization can rapidly adapt our processes and ways of working to deliver systems for small satellites.

Acknowledgments

I would like to thank my supervisors, friends, colleagues, and family for your continuous support and motivation. Thank you, Egil Eide, for motivating me, guiding me when needed, and providing valuable insight and ideas for research areas. Similarly, Tor Arne Johansen, for always taking the time to listen and discuss systems engineering and satellites. I am very grateful for having you two as supervisors, and I have learned a lot from you both personally and professionally.

I am fortunate to have been a part of the NTNU SmallSat Lab, where we built friendships, satellites, 3D-printed rockets and had a great time. I am grateful for all the discussions we have had together and the fun times. Thank you, Gara Quintana Diaz, Milica Orlandic, Mariusz E. Grøtte, Sivert Bakken, Alberto Dallolio, Joseph L. Garrett, Amund Gjersvik, Jon A. Justo, Bjørn A. Kristiansen, Dennis D. Langer, Fernando A. Agelet, Elizabeth F. Prentice, and Roger Birkeland. Special thanks to Roger for being a mentor and brainstormer and convincing me to apply for a PhD — it has been an incredible journey, and I am happy and proud of all the work we have done and the organization the team, and we have built. And, of course, thank you to all the students who have been a part of our satellite team since 2018 — without you, we would not have had a satellite!

To my brother-in-law Nikolai Holm, and my friend Jennifer Duggan, I needed to develop a transdisciplinary thesis with legs in many fields and to help me structure the work and my research method. Ditto for Knut Robert Fossum for doing parts of the research with me and acting as a mentor in Participatory Action Research and Project Management research. I am forever grateful for all the discussions I have had with Knut and look forward to new research projects and philosophical talks. To Aleksander L. Skrede and Øyvind Bjelland for all the fun discussions and brainstorming about agile processes.

To my dear friend and mentor Cecilia Haskins. I am so glad I met you! You have guided me throughout this process and pushed me when required or been a conversation partner when I needed that. Thank you for introducing me to INCOSE and the wonderful community of systems engineers around the world.

I would also like to acknowledge all the support I have received from the department. To Terje Mathiesen for all the help planning and setting up our lab that has been my home at NTNU. I also want to thank the great people in the Radio Systems and Circuit group. To all my colleagues and lunch-friends, Torbjörn Ekman, Jostein Grepstad, Reinold Ellingsen, Nina Lundberg, Astrid Aksnes, Jens Abraham, Asgeir Bjørgan, Silje Skeide Fuglerud, Fredrik Kjemperud Olsen, Einar Digernes, Hans Olaf Hågenvik, Ambjørn Dahle Bang, Erik Folven, Kristoffer Kjærnes, Sam Slotje, Anders Strømberg, Torstein Bolstad, Thomas Tybell, Erik Wessel-Berg and everyone else.

Next, to all my friends near and dear from everywhere, Yonsei University, NTNU, climbing, book club. Silje, Silje (yes, both of you), Ingebjørg, Inga, Christine, Vendi, Bojana, Anders, Anita, Ragnhild, Madeleine, Lizzy, Amarynth, Ann Christin, Sharm, Yeva, Ann, Daria.

To my dear family. First and foremost, my mother, Eva, for being a continuous inspiration in my life by being a role model with an impressive career and a big heart. You showed me that it was possible to be myself and achieve anything. I love our talks about management, projects, research, technology —– and the fact that your house is always open whenever I want to visit and bring both friends and colleagues. To Erik and Emma, my wonderful siblings. And all my aunts and uncles and cousins, with whom I love meeting and discussing and solving world problems. Thank you for inspiring me to be awesome. I am so grateful for your support and all the dinners we have had together in Trondheim throughout my PhD. Thank you for always being there and making me smile. My in-laws, Linn, Eva, and Hans Inge, for their curiosity and support through this journey.

Finally, dear Christer, thank you for being unconditionally supportive and generous. You never hold me back, and you are always there when I need you, with a smile and love. This is all I need.

Funding Information

This thesis work was carried out at the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU) in Trondheim, Norway, from November 2017 to May 2021. The primary supervisor was Associate Professor Egil Eide (NTNU), the co-supervisor was Professor Tor Arne Johansen (NTNU), and the supporting supervisor was Associate Professor Cecilia Haskins (NTNU). The work has been a part of the Small Satellite Laboratory at NTNU and supported through the funding from Norwegian Research Council grants no. 270959 and no. 223254. The Faculty of Information Technology and Electrical Engineering has funded this PhD position with strategic funds for small satellite development. The project has also received support from the Norwegian Space Agency.

Software tools used:

I acknowledge Vitech Corporation for the research licenses for CORE and GENESYS, Valispace for the academic license for using Valispace, and Sapienza Consulting for the educational license for using ECLIPSE. I would also like to thank Stephane Lecrampe at OBEO for support in using the Capella tool and Arcadia method. I thank Mark Simmons for all the support for using the CORE tool.

Supervision:

During the fellowship, I co-supervised the following master students: Anders Brørvik, Jhonattan Tolozo, Erik F. Trydal, Runar Gården Rovik, Eli Gjerde, Ingrid C. Jenssen, Martine Hjærtenes, Fredrik Hoel Bevreng, Sergio Farragut, Tord Hansen Kaasa, Henrik Galtung and Tuan Anh Tran. I also co-supervised the following bachelor students: Ada Kristine O. Moen, Eivind Sjøvold, Ola Jordheim, Jim Meløysund, Per Arne Johansen, Kristoffer Wood Finnson, Fredrik Nes Fredheim and Andreas Inderberg.

I served as project manager and systems engineer for the 6U-CubeSats HYPSO-1, and HYPSO-2 at the NTNU SmallSatLab from 2018–2021 for a team of 20–30 undergraduate and graduate students each year.

Acronyms

ADCS Attitude Determination and Control System.

ADGM Agile Decision-Guidance Method.

AIT Assembly, Integration, and Test.

AMOS Centre for Autonomous Marine Operations and Systems.

ASV Autonomous Surface Vehicles.

AUV Autonomous Underwater Vehicles.

BOB Break-Out Board.

CDR Critical Design Review.

COTS Commercial-Off-The-Shelf.

CPS Cyber-Physical Systems.

CS Constituent System.

CSRM CubeSat System Reference Model.

DE Digital Engineering.

DoD Department of Defense.

ECLIPSE ECSS Compliant Toolset for Information and Projects Support of Enterprises in Space.

ECSS European Cooperation for Space Standardization.

EPS Electrical Power Subsystem.

ESA European Space Agency.

FDIR Failure Detection, Isolation, and Recovery.

FMECA Failure Mode, Effects, and Criticality Analysis.

FTA Fault Tree Analysis.

GNSS Global Navigation Satellite System.

HIL Hardware-In-the-Loop.

HSI HyperSpectral Imager.

HYPSO HYPer-Spectral SmallSat for Ocean Observation.

IMU Inertial Measurement Unit.

INCOSE International Council on Systems Engineering.

IOD In-Orbit Demonstration.

IRL Integration Readiness Level.

ISO International Standards Organization.

ITU International Telecommunication Union.

LEO Low-Earth Orbit.

MASSIVE Mission-oriented Autonomous Systems with Small Satellites for Maritime Sensing, Surveillance and Communication.

MBSE Model-Based Systems Engineering.

MCE Model-Centric Engineering.

MDR Mission Design Review.

MIT Massachusetts Institute of Technology.

MVP Minimum Viable Product.

NASA National Aeronautics and Space Administration.

NTNU Norwegian University of Science and Technology.

OBC On Board Computer.

OMG Object Management Group.

OODA Observe-Orient-Decide-Act.

PDR Preliminary Design Review.

PM Project Management.

PMI Project Management Institute.

RAM Reliability, Availability, Maintainability.

RCN Research Council of Norway.

RGB Red Green Blue.

RID Review Item Discrepancy.

SE Systems Engineering.

SERC Systems Engineering Research Center.

SoS System of Systems.

SPADE Stakeholders, Problem, Alternatives, Decision-making, Evaluation.

SRL System Readiness Level.

SSM Soft Systems Methodology.

SSWG Space Systems Working Group.

TRL Technology Readiness Level.

UAV Unmanned Aerial Vehicles.

Definition of Terms

The following definitions are applied in this work, separated into "general terms" and project-specific "satellite terms."

General Terms

Constituent system: From ISO21839 [2]: "Constituent systems can be part of one or more SoS. Note: Each constituent is a useful system by itself, having its own development, management goals, and resources, but interacts within the SoS to provide the unique capability of the SoS."

Cube Satellite (CubeSat): A satellite built according to the CubeSat Design Specification [3]. CubeSats consist of $10\mathrm{cm} \times 10\mathrm{cm} \times 10\mathrm{cm}$ units (U), where each unit weighs less than $1.3\mathrm{kg}$. The CubeSats can be of different sizes, for example 1U, 3U, or 6U, see Figure A.

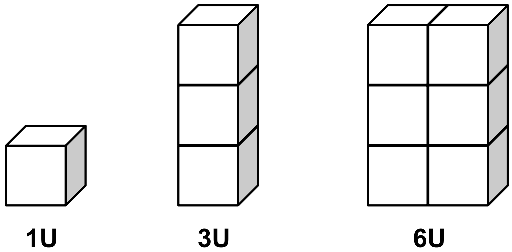


Figure A. Examples of CubeSat configurations.

Dispersed team: When team workers are spread out over a large area.

Distributed team: For when there are both workers on-site in the office and workers in their home offices.

Emergence: That the behavior of a system cannot be described wholly by the behavior of its parts [4], which also is applicable for System-of-Systems.

In-situ: Sampling on-site/on-location. For oceanography, this means taking samples of the water or measurements in the water, as opposed to observing from a distance.

Modes of communication: The modes of communication include "interpretative" (one-way communication where the listener might not understand everything that is said but can interpret the whole meaning from the context), "interpersonal" (two-way communication), and "presentational" (also one-way, but typically as a presentation which could be rehearsed or recorded). Furthermore, how the communication is transmitted "verbal", "non-verbal" (body language and gestures), "visual", and "written."

Modes of interaction: The modes of interaction typically involve "deciding" (needing to exchange the right types of information and make decisions), "clarifying" (directing towards a shared understanding), "considering" (involves active and independent thought), "reading" (in-depth finding which can involve some considering) and "finding" (just finding the information and not considering it).

Project: According to the Project Management Body of Knowledge [5]: "A project is a temporary endeavor undertaken to create a unique product, service, or result."

Project Management: According to the Project Management Body of Knowledge [5]: "Project management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements."

Nanosatellite: A spacecraft of less than 10kg [6].

Research: According to OECD [7]: “Any creative systematic activity undertaken in order to increase the stock of knowledge, including knowledge of man, culture and society, and the use of this knowledge to devise new applications.”

Small satellite: A spacecraft of less than 1000kg [6].

System context: The external elements that interact with the system-of-interest, such as environment or actors, which are needed to define to understand the system and its purpose as a whole.

System-of-interest: From ISO15288 [8]: “The system whose lifecycle is under consideration.”

System-of-Systems: A System-of-Systems is a collection of systems that maintain their operational and managerial independence [9].

Systems Engineering: From Systems Engineering Book of Knowledge [10]: “A transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods.”

Virtual teams: A particular case of a *distributed team*, where the players are spread out geographically and organizationally but are linked through technology and a common project [11].

Satellite Terms

Since the main case study has been the HYPer-Spectral SmallSat for Ocean Observation (HYPSO) mission, introducing some of the terms used in this thesis relevant to the satellite is included.

ADCS: Attitude Determination and Control System (ADCS) is the system that monitors and controls the movement and attitude of the satellite. Some of the algorithms used by the HYPSO mission are developed in collaboration between Norwegian University of Science and Technology (NTNU) and NanoAvionics.

Frame subsystem: The frame subsystem consists of the mechanical support structures and frame, which binds the satellite together.

HSI: HSI means Hyperspectral Imager. It consists of mechanics, optics and a detector that can detect more than 20 spectral bands.

HYPSO: HYPerspectral small Satellite for Oceanographic observations. The 6U CubeSat built by the project team.

HYPSO team/organization: The HYPSO team or organization refers to the team concerned with developing the satellite.

OBC: The On Board Computer (OBC) is the satellite bus computer, which some refer to as flight computer.

Payload controller: The payload controller (PC) acts as an interface between the payload (developed by NTNU) and the satellite bus (developed by NanoAvionics).

Payload subsystem: The payload subsystem is the system that interacts with the subject of interest (in the HYPSO case, the ocean).

PicoBOB: The PicoBOB is the payload processing system. It consists of an in-house developed interface board, Break-Out Board (BOB); a PicoZed processing board from AVNET corporations, and shield plates to protect the electronics from radiation.

Power subsystem: The power subsystem (also called Electrical Power System (EPS)) makes up the electrical power supply, the battery packs, and the solar panels on the CubeSat.

RGB camera: The Red Green Blue (RGB) camera is a part of the payload to facilitate georeferencing and pixel registration.

Thermal subsystem: The thermal subsystem can be passive or active and is responsible for ensuring that the components do not overheat or get too cold during operation.

Contents

List of Contributions

Paper A
Honoré-Livermore, Evelyn, "CubeSats in University: Using Systems Engineering Tools to Improve Reviews and Knowledge Management," Conference on Systems Engineering Research, *Procedia Computer Science*, Volume 153, pp. 63–70, 2019 [12]. `https://doi.org/10.1016/j.procs.2019.05.056`.

E.H-L. wrote and prepared the paper. Cecilia Haskins and Egil Eide supervised the work and reviewed the paper.

Paper B
Honoré-Livermore, Evelyn; Haskins, Cecilia, "Model-Based Systems Engineering for CubeSat FMECA," In: Madni, A.M., Boehm, B., Erwin, D., Moghaddam, M., Sievers, M., Wheaton, M., (eds) Recent Trends and Advances in Model Based Systems Engineering. Conference on Systems Engineering Research, Springer, Cham. [13].

Both authors conceptualized the paper. E.H-L. wrote and prepared the original draft. Both authors reviewed the original draft and contributed to the revision of the paper.

Paper C
Honoré-Livermore, Evelyn; Bakken, Sivert; Prentice, Elizabeth Frances, "Factors Influencing the Development Time from TRL4 to TRL8 for CubeSat Subsystems at a University," 9th International Systems & Concurrent Engineering for Space Applications Conference (2020) (virtual) [14].

All authors conceptualized the paper. E.H-L. wrote and prepared the original draft. S.B. wrote the code to analyze the system, E.H-L. performed the interviews. S.B. wrote the section on software team development, E.F.P wrote the section on hardware team development. All authors analyzed the results and wrote the discussion together.

Paper D

Honoré-Livermore, Evelyn; Birkeland, Roger, "Managing Product Development and Integration of a University CubeSat in a Locked down World," 2021 IEEE Aerospace Conference (50100), 2021, pp. 1-12, `http://dx.doi.org/10.1109/AERO50100.2021.9438490` [15].

Both authors conceptualized the paper. E.H-L. wrote and prepared the original draft. Both authors reviewed the original draft and contributed to the revision of the paper. Cecilia Haskins supervised the work and reviewed the paper.

Paper E

Honoré-Livermore, Evelyn; Birkeland, Roger; Bakken, Sivert; Garrett, Joseph L.; Haskins, Cecilia, "Digital Engineering Management in an Academic CubeSat Project," Special Issue on Systems Engineering Challenges in Journal of Aerospace Information Systems (2021) [16]. `https://doi.org/10.2514/1.I010981`.

All authors conceptualized the paper. E.H-L. wrote and prepared the initial outline and finalized the submission. S.B., R.B. and J.L.G. all contributed to background, results and discussion sections. C.H. reviewed the draft and contributed to final submission.

Paper F

Honoré-Livermore, Evelyn; Lyells, Ron; Garrett, Joseph L.; Angier, Robert (Rock); Epps, Bob, "An Agile Systems Engineering Analysis of a University-built CubeSat," IS2021 INCOSE International Symposium on Systems Engineering, 31, pp. 1334–1348 [17]. `https://doi.org/10.1002/j.2334-5837.2021.00904.x`

E.H-L. and R.L. conceptualized the paper. R.L, R.A. and B.E. developed the method. All authors contributed to data collection and review of the paper.

Paper G

Honoré-Livermore, Evelyn; Birkeland, Roger; Haskins, Cecilia, "Addressing the Sustainable Development Goals with a System-of-Systems for Monitoring Arctic Coastal Regions," INCOSE International Symposium (ISSN 2334-5837), Vol. 30, no. 1, pp. 604–619, 2020 [18]. `http://dx.doi.org/10.1002/j.2334-5837.2020.00743.x`

All authors conceptualized the paper. E.H-L. selected the framework for the stakeholder analysis and SoS analysis, performed together with R.B.. E.H-L. and R.B. wrote and prepared the original draft. All authors reviewed the original draft and contributed to the revision of the paper.

Paper H

Honoré-Livermore, Evelyn; Dallolio, Alberto; Birkeland, Roger; Langer, Dennis D.; Haskins, Cecilia; Johansen, Tor Arne, "MBSE modeling of a SoS with a small satellite and autonomous surface vessels for persistent coastal monitoring," 2021 16th International Conference of System of Systems Engineering (SoSE), 2021, pp. 156-161, `http://dx.doi.org/10.1109/SOSE52739.2021.9497470` [19].

All authors conceptualized the paper. E.H-L. did the modeling in Capella, and all authors contributed to the modeling activities through discussions and revisions. E.H-L. wrote and prepared the original draft. All authors reviewed the original draft and contributed to the revision of the paper.

Paper I

Honoré-Livermore, Evelyn; Fossum, Knut Robert; Veitch, Erik, "Academics' Perception of Systems Engineering and Applied Research Projects," Systems Engineering, (The Journal of The International Council on Systems Engineering), Wiley Periodicals, 2021, pp. 1–16, `http://dx.doi.org/10.1002/sys.21599` [20].

All authors conceptualized the paper. E.H-L. and K.R.F. performed the data collection. All authors analyzed the data. E.V. did the statistical analysis, wrote the statistical methods' section, and made the correlation figure. E.H-L. and K.R.F. wrote the original draft. All authors contributed to revision.

Paper J

Fossum, Knut Robert; Honoré-Livermore, Evelyn; Veitch, Erik; Palmer, Erika; Haskins, Cecilia "Towards an integrated project complexity narrative – A case study of academic organizations," Accepted to be published in Systems Engineering, (The Journal of The International Council on Systems Engineering), Wiley Periodicals, (2022) [21].

All authors conceptualized the paper. E.H-L. and K.R.F. performed the data collection. E.H-L., K.R.F., and E.V. analyzed the data. K.R.F. wrote most of the paper in discussions with E.H-L and E.P. E.P and E.H-L. wrote the section on SE for complexity. All authors contributed to revision.

Other Results

Other results and conference contributions during the doctoral studies not included in the thesis:

1. Honoré-Livermore, Evelyn; Eide, Egil, "Survey of Lean Satellite Development Management Software," Small Satellites Systems and Services – The 4S Symposium, Sorrento, Italy (2018).

2. Honoré-Livermore, Evelyn, "Survey of Lean Satellite Development Management Software," International Workshop on Lean Satellite, Kyushu, Japan (2018).

3. Honoré-Livermore, Evelyn, “HYPSO - protecting the oceans,” International Workshop on Lean Satellite, Tokyo, Japan (2019).

4. Quintana Díaz, Gara; Birkeland, Roger; Honoré-Livermore, Evelyn; Ekman, Torbjörn, “An SDR mission measuring UHF signal propagation and interference between small satellites in LEO and Arctic sensors,” 33rd Annual AIAA/USU Conference on Small Satellites, Logan, UT, USA (2019).

5. Honoré-Livermore, Evelyn, “Information Sharing and Structuring for Development and Integration of CubeSats,” International Workshop on Lean Satellite, (2020) (virtual).

6. Larsen Skrede, Aleksander; Bjelland, Øystein; Honoré-Livermore, Evelyn, “Work-in-Progress: An agile approach to formative assessment in higher education,” 2021 IEEE Global Engineering Education Conference (EDUCON), 2021, pp. 1126-1130, `http://dx.doi.org/10.1109/EDUCON46332.2021.9454060`.

7. Grøtte, Mariusz E.; Birkeland, Roger; Honoré-Livermore, Evelyn; Bakken, Sivert; Garrett, Joseph L.; Prentice, Elizabeth F.; Sigernes, Fred; Orlandic, Milica; Gravdahl, Jan Tommy; Johansen, Tor Arne, “Ocean Color Hyperspectral Remote Sensing with High Resolution and Low Latency - the HYPSO-1 CubeSat Mission,”, pp. 1-19, IEEE Trans. on Geoscience and Remote Sensing, (2021). `https://doi.org/10.1109/TGRS.2021.3080175`.

8. Honoré-Livermore, Evelyn; Garrett, Joseph L.; Lyells, Ron; Angier, Robert (Rock); Epps, Bob, “An Agile Systems Engineering Analysis of Socio-technical Aspects of a University-built CubeSat,” INSIGHT Magazine Article, Accepted to be published 2021.

9. Dallolio, Alberto; Quintana-Diaz, Gara; Honoré-Livermore, Evelyn; Garrett, Joseph; Birkeland, Roger; Johansen, Tor Arne, “A Satellite-USV System for Persistent Observation of Mesoscale Oceanographic Phenomena,” Special Issue on Satellite Communications in Remote Sensing, 13(16), 3229. Retrieved from `https://www.mdpi.com/2072-4292/13/16/3229`

10. Bakken, Sivert; Honoré-Livermore, Evelyn; Birkeland, Roger; Orlandic, Milica; Prentice, Elizabeth F.; Garrett, Joseph; Langer, Dennis D.; Haskins, Cecilia; Johansen, Tor Arne, “Software Development and Integration of a Hyperspectral Imaging Payload for HYPSO-1,” SICE 2022 International Symposium on System Integration.

11. Bakken, Sivert; Birkeland, Roger; Garrett, Joseph; Marton, Per A.R.; Orlandic, Milica; Honoré-Livermore, Evelyn; Langer, Dennis D.; Haskins, Cecilia; Johansen, Tor Arne, “Testing of Software-Intensive Hyperspectral Imaging Payload for the HYPSO-1 Satellite” SICE 2022 International Symposium on System Integration.

12. Prentice, Elizabeth F.; Honoré-Livermore, Evelyn; Bakken, Sivert; Birkeland, Roger; Henriksen, Marie B.; Gjersvik, Amund; Hjærtenes, Martine; Johansen, Tor Arne; Agelet, Fernando Aguado;

Medina, Fermín Navarro, “Assembly, Integration, and Testing Strategy of a Hyperspectral Imaging CubeSat, HYPSO-1,” To be submitted to Acta Astronautica.

13. Garrett, Joseph L.; Bakken, Sivert; Prentice, Elizabeth F.; Langer, Dennis D.; Leira, Fredrik; Honoré-Livermore, Evelyn; Birkeland, Roger; Grøtte, Mariusz A.; Johansen, Tor Arne; Orlandić, Milica, “Hyperspectral Image Processing Pipelines on Multiple Platforms for Coordinated Oceanographic Observation,” 11th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2021.

Chapter 1

Introduction

Don't let anyone rob you of your imagination, your creativity, or your curiosity. It's your place in the world; it's your life. Go on and do all you can with it, and make it the life you want to live.

Mae Jamieson

Small satellites have offered, since the late 1990s, a new approach to space science, communication, earth observation, and education. This has been facilitated by the proliferation and miniaturization of low-cost electronics which have lowered the barriers to entry in tandem with increasing launch opportunities [6]. While access to space was previously limited to the governments of highly industrialized nations, we now see that small companies, poor and middle income countries, and even universities are developing capabilities to build satellites that deliver scientific results and technological breakthroughs [22].

However, there are still many failed missions, or missions that are not delivering to their full potential or meeting initial schedule or cost targets [23–25]. University CubeSat teams struggle with high turnover, knowledge management, and the balance of coursework and satellite tasks [26–28]. This research investigates how an integrated approach to agile Systems Engineering and Project Management can support the development processes, in the context of applied research projects.

In this chapter, I introduce the problem and thesis. A more thorough explanation of terms used in the thesis is given in subsequent chapters. First, I describe the problem statement and the research objectives. Next, I summarize my contributions to this research. Finally, I present an outline of the thesis to support the readers through the narrative.

1.1 Ocean Observation

The backdrop for the research in this thesis has been the HYPSO CubeSat mission, which is funded through the MASSIVE project. Therefore, an explanation of the project and mission follows to provide context to the research objectives. The HYPSO project has also been the main case study for the research and the publications.

The NTNU Center for Autonomous Marine Operations and Systems (AMOS) was established with a vision in mind, of using autonomous systems in concert with manned systems to provide responsive ocean monitoring in harsh oceanic environments [29]. NTNU's strategic initiative on ocean monitoring spans multiple departments, including the Department of Engineering Cybernetics, Department of Electronic Systems, and the Department of Marine Technology. The MASSIVE project is one of many research projects associated with this goal, effectively being a part of a "portfolio-of-systems" collectively managed by NTNU [30].

1.1.1 MASSIVE Project

The MASSIVE project is a part of a larger strategic initiative at NTNU for building cost-effective solutions for observing the impact of the changing climate on the world's oceans [29]. As part of the increasing trend for earth observation from space, there is a drive for better oceanographic monitoring, especially in remote environments [31–33]. Conventional methods using large international surveillance satellites and ship-based surveillance are expensive and require extensive planning which is not optimal for observing the rapid changes in the ocean [34]. Coordinated near real-time data from ground-based and aerial-based sensor systems are needed to provide a comprehensive picture of oceanographic phenomena [35]. The MASSIVE project supports the development, manufacturing, and launch of two satellites, financing PhD candidates and includes coordinating operations between satellites and autonomous vehicles. The plan is to launch two small satellites within a short time period with novel on-board processing of hyperspectral data from an electro-optical payload developed in-house. These small satellites will provide hyperspectral information for e.g., scientists wanting raw data, or fish farmers wanting operational data that can be acted upon. The data will be collected in a greater repository with measurements collected by other assets or satellites, such as water temperature, salinity, or other ocean color data. This data can, with appropriate processing, become actionable information. Together, this can provide decision-makers with improved situational awareness, in near real-time, of the state of the ocean. The project proposes creating a network of small satellites [36], Autonomous Underwater Vehicles (AUV), Autonomous Surface Vehicles (ASV), and Unmanned Aerial Vehicles (UAV) in concert with conventional vehicles, buoys, and fixed sensor networks, as illustrated in Figures 1.1 and 1.2, to meet the needs of the scientific community.

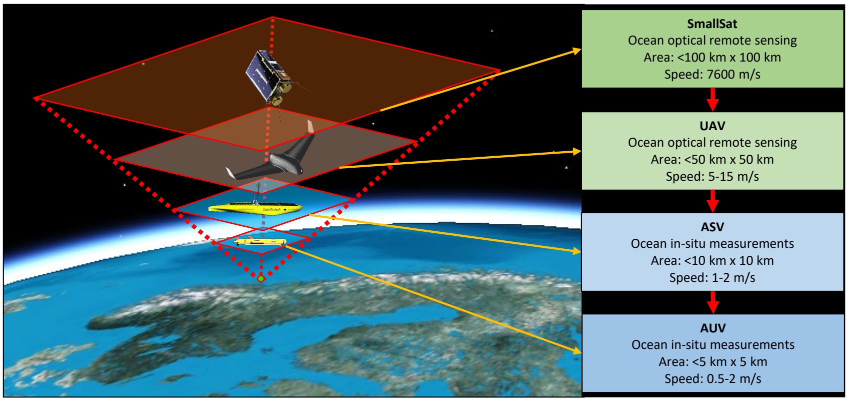


Figure 1.1: Observational pyramid for oceanographic measurements. Figure from Mariusz E. Grøtte.

There has been an increase in using multi-agent (i.e. different types of assets) autonomous vehicles for oceanographic observations [37–41], often gathering researchers from different institutions and employing assets that have not operated together previously. For example, by deploying constellations and by combining satellites from different operators and providers [42], this can provide better situational awareness.

Operating multiple assets simultaneously to deliver the required information needs to take resilience and performance into account, and allow for dynamic allocation of functionality. This can increase the operational coverage and utility, and go beyond what individual Constituent System (CS) (i.e. an individual system participating in a System of Systems (SoS)) can deliver on their own [42–44]. Combining the different assets can provide better resilience towards varying environmental conditions, where the SoS can allocate which systems to deploy. For example, when there is high cloud coverage (when satellites provide little information) and low-flying UAVs or surface vehicles would provide more data, or combining satellite weather data with in-situ knowledge to plan missions. Another opportunity is distributed computing, where the SoS can trade e.g., power, data budgets, communication links and latency requirements to distribute where computation should happen amongst the assets. The dynamics of the environments and assets can produce emergent behavior which the SoS must react and respond to, so that it can provide consistent performance and capabilities to the stakeholders [42, 45].

The complete proposed system can provide substantially more continuous information about observed targets and features of scientific interest, and do so synoptically. A key contributor in this observational pyramid, shown in Figure 1.1, is the space segment, which for MASSIVE is supported by small satellites from the HYPSO project. The project is highly multidisciplinary, involving technologies and people from cybernetics and control studies, electronics and embedded design, firmware programming, optical design, mechanical and electrical design, project management, systems engineering, operations, and product development and design studies.

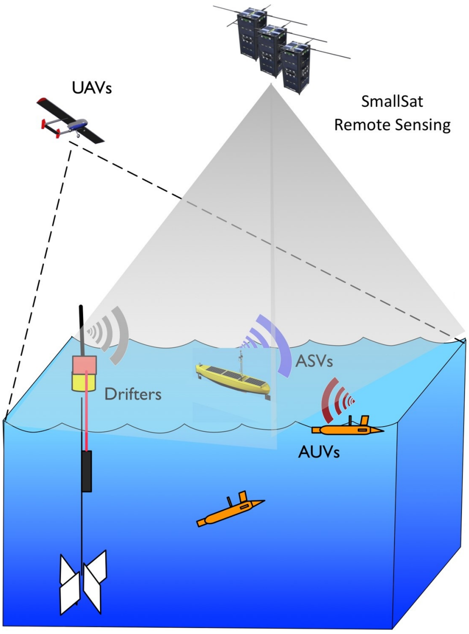


Figure 1.2: Overview of multiple assets observing the same area in the ocean. The small satellites and UAVs can be equipped with hyperspectral cameras, and the other assets can provide in-situ measurements. Figure from Mariusz E. Grøtte.

One of the research objectives in this PhD is to contribute to the development of the MASSIVE SoS, and establish an understanding of HYPSO's role and capabilities in this context. Figure 1.2 provides an overview of assets in the MASSIVE project, and how they observe the same area of interest. The sky- and space-borne assets in Figure 1.2 can be equipped with hyperspectral imaging sensors to provide high spectral and spatial resolution data. The surface vehicles can provide information about wind and weather, water temperature, and upper-water column in-situ measurements. The underwater vehicles can provide in-situ measurements at lower depths, and more information about currents which complements the data collection and can support predictive ocean modeling.

1.1.2 HYPSO Project

The HYPSO project objective is to build and launch two small satellites with an in-house developed HyperSpectral Imager (HSI) payload [34]. The satellites are named HYPSO-1 and HYPSO-2. The HYPSO project started in 2017 and is expected to launch the first satellite in December 2021. The NTNU Small Satellite Lab is responsible for the project. A project team of 6–8 PhD candidates and 20–30 BSc/MSc students (per year) have been responsible for both mission, the HSI payload, and the associated ground segment. The primary mission objective (MO) and secondary mission objectives (SMO) are [34]:

- **MO-001:** To provide and support ocean color mapping through a Hyperspectral Imager (HSI) payload, autonomously processed data, and on-demand autonomous communications in a concert of robotic agents at the Norwegian coast.
- **SMO-001:** To collect ocean color data and to detect and characterize spatial extent of algal blooms, measure primary productivity using emittance from fluorescence-generating micro-organisms and other substances resulting from aquatic habitats and pollution to support environmental monitoring, climate research, and marine resource management.
- **SMO-002:** Enhance autonomous AI-based coordinated operation of satellites and in-situ robotic platforms in the ocean as a robotic network.
- **SMO-003:** Collect oceanographic community input and feedback on key science drivers for coordinated space, aerial, surface, and underwater observations for targeted bio-geochemical processes (e.g., Harmful Algal Blooms (HABs), oceanic fronts, internal waves).

Figure 1.3 shows a model of the satellite. The satellite bus is provided by NanoAvionics Ltd., and the payload is developed and manufactured by NTNU. The telescope is the HSI component shown in Figure 1.4, and details about the camera and its development can be found in Prentice et al. [46]. The main challenges for the payload development have been incorporating Commercial-Off-The-Shelf (COTS) components, tailoring them for space, and ensuring the high performance needed.

Hyperspectral Imagers generate large amounts of data. Depending on what the end-user needs are, the operators can command the satellite to deliver different data products. The operators in the ground station control this by changing the payload processing modes, and the processing pipeline can be updated in-flight [34]. The payload has its own processing unit, with software and operating system have been developed in-house by NTNU [47].

A more in-depth description of the HYPSO project is given in Section 4.6. An overview of CubeSat subsystems is given in Figure 1.5, and the payload building blocks in Figure 1.6. The HYPSO CubeSat consists of the following subsystems: a communication subsystem; a power subsystem (referred to as the Electrical Power Subsystem (EPS)) which includes solar panels, batteries, and a power distribution board; an ADCS subsystem which includes the Inertial Measurement Unit (IMU), Global Navigation Satellite System (GNSS), star-tracker, magnetorquers and magnetometers — which together are used to determine and control the position and attitude of the spacecraft; the payload controller subsystem which handles dataflow to and from the payload; the OBC subsystem which acts as the spacecraft computer, responsible for all the other subsystems management; the thermal subsystem which consists of active and passive thermal elements to ensure spacecraft performance in a harsh environment; the frame subsystem supporting all the hardware; and the payload subsystem. The payload subsystem, shown in Figure 1.6, developed and built by the HYPSO project team includes a processing unit, a hyperspectral camera, a RGB camera, and hardware (harness and frame) necessary to support it.

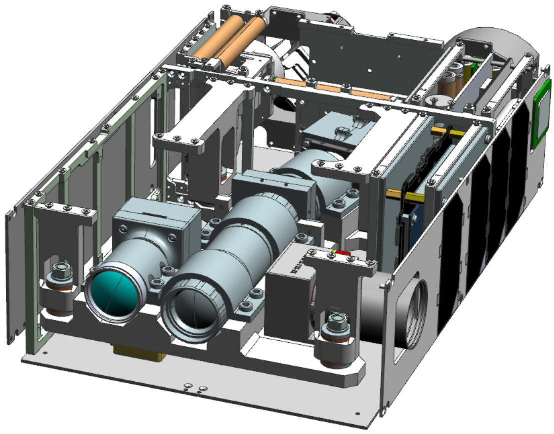


Figure 1.3: Model of the HYPSO-1 satellite bus and payload (center). The envelope of the CubeSat is 20 cm by 10 cm by 30 cm. It is equipped with an RGB camera and a star-tracker to better image registration. More details about the mission can be found in [34]. Figure from Elizabeth F. Prentice.

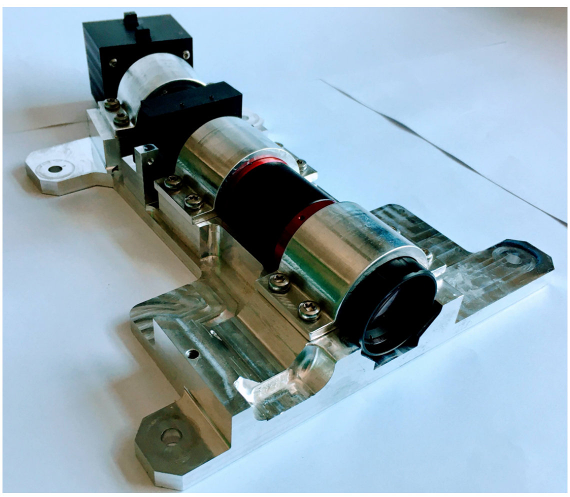


Figure 1.4: Picture of the Hyperspectral Imaging payload hardware prototype. Picture from Elizabeth F. Prentice.

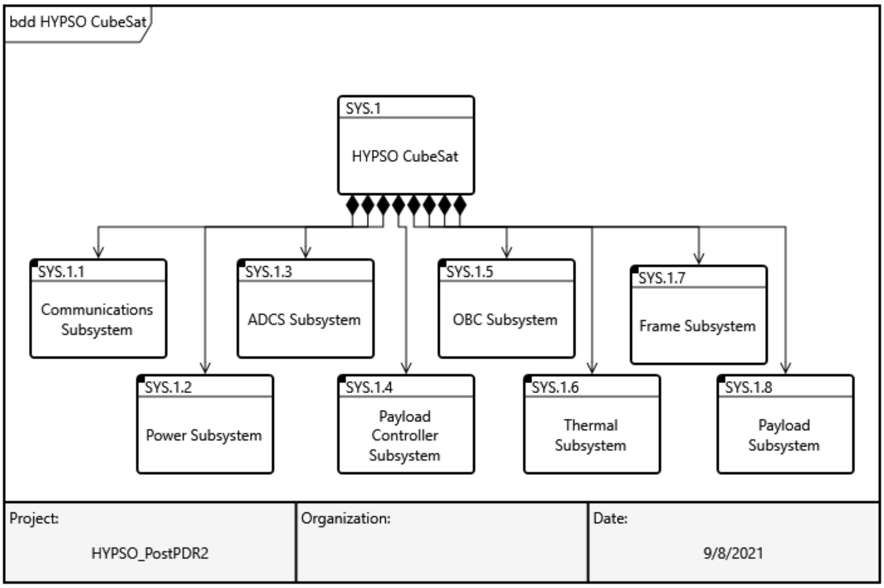


Figure 1.5: Overview of HYPSO-1 CubeSat. Diagram made using GENESYS by Vitech corp.

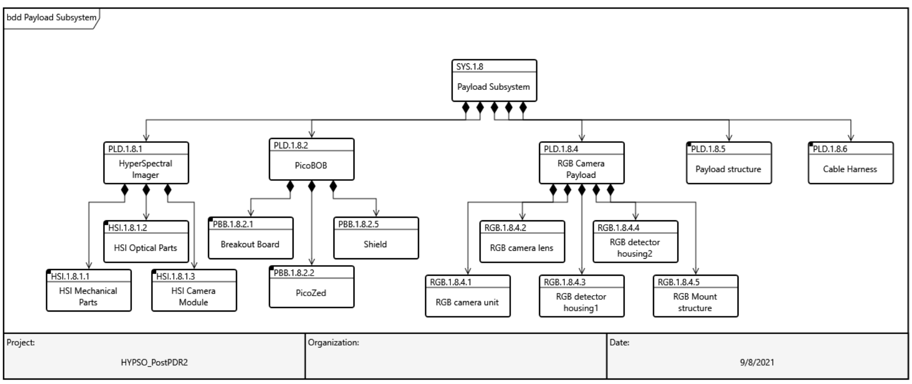


Figure 1.6: Overview of the HYPSO-1 payload. Diagram made using GENESYS by Vitech corp.

1.2 Building Scientific Research Satellites at Universities

This section summarized findings that motivated (M) the eventual research directions for the this thesis. Universities have several purposes, and among their main objectives are education and contributing to cutting-edge research. The "projectification" of research in universities, combined with more applied research activities, creates challenges for the traditional research approaches [48, 49]. Project-based research creates an increased need for advanced infrastructure and multidisciplinary collaborations. For example, when the research needs algorithmic development, simulation, and implementation and testing of these. Students and PhD researchers are encouraged to participate in these development activities and find it challenging to balance their academic responsibilities with research goals and engineering work [50].

M1. The increased projectification of research in universities requires new capabilities of the organization for managing projects to deliver highly complex systems concurrently with other university duties.

The CubeSat standard for small satellites coupled with the increased availability and capabilities of microelectronics enable universities to design, develop, and integrate satellite payloads and spacecraft systems at relatively low costs [6, 51]. The university-built CubeSats are increasingly used for scientific purposes but still involve a great deal of "supportive engineering" work. In comparison to the traditional space projects, the CubeSat projects take higher risks. Traditional satellite projects may develop over ten years and cost more than €100 million. Small satellite projects on average last three years and can cost around €1-€2 million because of lower material costs, shorter development time, and lower launch costs. In reducing time it is a challenge to maintain low risk and low cost. Low risk, in this context, means agreeing upon the minimum success criteria defined in collaboration between the project team and end-users. Low cost often means using commercial products not fully certified for space.

While there are well-established standards (e.g., European Cooperation for Space Standardization (ECSS), International Standards Organization (ISO)) or military specifications and processes for developing "traditional" satellites, such as the big projects from the European Space Agency (ESA) and National Aeronautics and Space Administration (NASA), similar references for small satellites are not readily available [22, 52]. The existing body of standards amounting to several hundreds of pages and may seem overwhelming for university CubeSat projects to navigate. Some standards such as the CubeSat 101 standard from NASA [53], the ISO lean satellite standard [54], and the ECSS CubeSat standard [55], have been established in the later years. Moreover, many "lessons learned" papers are published each year at several conferences [26,27,51,56–58], providing valuable insights for new university CubeSat projects.

Many of the resources on a university projects are students who remain involved for only one semester or one year. The high turnover within a project team makes knowledge management critical, and necessitates good on-boarding and off-boarding processes with a well-designed information system. Previous research suggests that there are several challenges when conducting CubeSat projects at universities: project management and team structure [22, 59–61]; balancing school work and satellite building [60]; ensuring momentum [59, 60]; and ensuring the project success [22, 59]. Furthermore, the students may not have prior experience in aerospace engineering or project work and are only temporarily associated with the project due to the academic calendar and eventual graduation. The literature also reports a large percentage of failed university CubeSat missions [23, 25], where the lack of system-level testing and ad-hoc processes have been identified as potential culprits. In summary, studies recommended the following:

1. Secure funding sources for the length of the project.
2. Interface control is a key factor to a successful multidisciplinary product team.
3. Instill a sense of ownership for the project success and conduct continuous team building activities.
4. Schedule the project so that exam periods and holidays are respected.
5. Establish a process for knowledge transfer.
6. Manage expectations of stakeholders: students, supervisors, and external actors.

Model-Based Systems Engineering (MBSE) offers a paradigm-shift in Systems Engineering (SE), and may improve the situation in terms of managing requirements, knowledge, interfaces, verification and validation activities [62–67]. MBSE can also assist in trade space exploration [68, 69]. Using a model enables easier communication of architecture choices between affected subsystems. For example, how different image sensors affect the power budget, the ground footprint (the area on ground imaged), the data link between spacecraft and ground segment, etc. The International Council on Systems Engineering (INCOSE) Space Systems Working Group has developed a CubeSat System Reference Model (CSRM) as a platform-independent and SysML-compliant model [70]. The purpose of the model is to enable both universities, government entities, and industries to develop CubeSats systematically. Additionally, MBSE offers possibilities for reuse of model elements and relationships, which is of value for MASSIVE and other CubeSat projects that want to build multiple satellites.

M2. The challenges university CubeSat projects face are sociotechnical in nature, and require an approach that looks at both the technical factors (i.e., the product challenges, e.g., software bugs, inexperience, interface mismatch, poor design choices) and the managerial responsibilities (i.e., project planning, teamwork, resource allocation, knowledge management, stakeholder management) in tandem.

As more scientific tasks are assigned to university CubeSats, the CubeSats are conceived to collaborate in scientific networks with other assets, such as unmanned aerial vehicles [71], land sensors systems [72], and integrated with communication systems. These networks may be maintained and operated by many different organizations, suggest the value an SoS [9] approach for analyses to ensure that the networks can deliver the capabilities the end-user needs [19, 35]. The integration of the CubeSat as one of many constituent systems into an SoS can be challenging, not only because of interface management across organizations, but also concerning reliability, resilience, stakeholder management, emergence, and verification and validation [73–75].

M3. There is a need for integrating different types of systems to deliver situational awareness capabilities for ocean monitoring. System integration of legacy systems, autonomous vehicles, small satellites and communication services presents significant challenges.

Moreover, the MASSIVE project is continuously evolving with new assets added or new services required as new understanding of the available technologies and scientific needs evolves. This results in a dynamic stakeholder picture with changing requirements, and the traditional plan-driven methodologies such as the waterfall model or the vee-model are considered inappropriate [76–78] for managing the university CubeSat project HYPSO. Agile practices that have gained traction in the software development the past 20 years [79, 80] offer techniques for managing projects with changing requirements and many unknowns, but presents challenges for hardware projects [81–83].

M4. The traditional plan-driven methodologies are not well-suited for managing neither university CubeSat projects with changing schedules, nor delivering systems with changing requirements and capability needs.

These challenges and recommendations, and the MASSIVE project needs motivate the research objective and related questions.

1.3 Research Objective

This research focuses on university-led small satellite development projects, and the understanding of what can increase the success rate of university-developed small satellites with scientific payloads, especially when developed for use in a greater system-of-systems context together with other autonomous systems. The objective of this PhD research is to generate knowledge to support the understanding of applied research projects in academia, specifically projects related to space systems. This knowledge can be applied to create a process with supporting methods to *reduce the time between mission concept definition and launch while still ensuring project success* through the application of SE and PM tools and practices. My research explores the integration of agile Systems Engineering and Project Management tailored to a university environment of space engineering to support the development of small satellites and to manage applied research projects.

To effectively perform the work to integrate systems, we can structure our understanding of the development environment using Martin's (2004) approach. First, we recognized the Context of our problem domain as the scientific communities behind MASSIVE. Then, we need to understand what makes the Realization System (the project team) developing the Intervention System (the small satellite) effective [84]. Furthermore, if we want to integrate multiple Collaborating systems, or operate the eventually Deployed systems using ground stations (Sustainment systems), we need to consider the value of the SoS viewpoint. Together, these contribute to a better understanding of how an integrated approach to SE and PM can be utilized to achieve a successful project.

To focus the research, the following research questions were addressed. By answering these questions, the research objective can be met. The development of research questions have been iterative in nature, refined based on the learning and insights gained through the research.

RQ-0: What are known sociotechnical challenges in university CubeSat projects and how can they be addressed?

Understanding the problem space and known challenges for university CubeSat projects is the basis for this thesis. There are different cultural contexts for all projects, the organization forms differ, and generalizing lessons learned or problems is difficult. However, this basic research question will identify and summarize known challenges based on the literature review and the HYPSO project case study.

Based on the findings from RQ-0, the following research questions are formulated, see Figure 1.7.

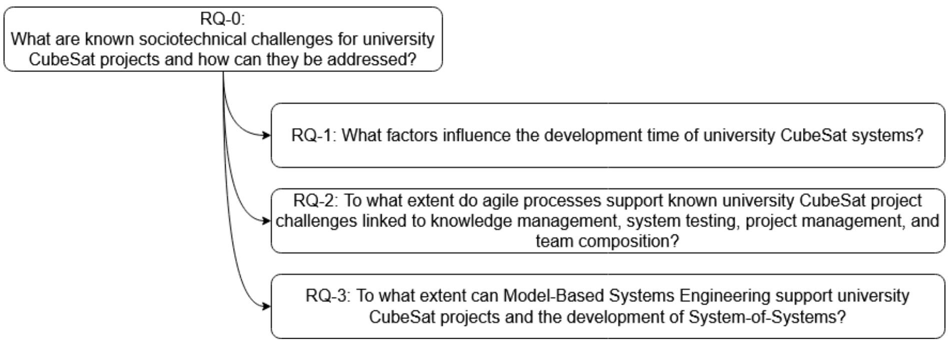


Figure 1.7: The findings from RQ-0 provide the foundation for RQ-1, -2, and -3.

RQ-1: What factors influence the development time of university CubeSat systems?

This research question aims at building an understanding of what factors influence the development time of university CubeSats, focusing on the HYPSO project. The objective is to enable improvement of the development time to achieve the overarching goal of rapid delivery of integrated small satellites in a university context to reduce th etime between ideation and launch. However, to improve, we must first understand what factors can be changed.

RQ-2: To what extent do agile processes support known university CubeSat project challenges linked to knowledge management, system testing, project management, and team composition?

Based on the literature review and characterization of the problem space, an agile approach was chosen for managing the HYPSO project. This research question aims at evaluating to what extent agile practices address the challenges found in RQ-0.

RQ-3: To what extent can Model-Based Systems Engineering support university CubeSat projects and the development of System-of-Systems?

Based on the literature review and the characterization of the problem space, MBSE has been identified as a useful paradigm for providing an integrated traceable description of the system-of-interest. Moreover, research indicates that MBSE is appropriate for SoS development. However, MBSE comes with challenges in training and the validity of the models, and it is not clear if there is a return-on-investment of time spent modeling for university CubeSat projects. This research question aims to build knowledge on how MBSE can support university CubeSat projects and their known challenges identified in RQ-0, based on the HYPSO project case study and its role in the greater ocean observational pyramid.

1.4 Research Contributions

First, this thesis contributes to our understanding of applied research projects in academia through the longitudinal replication case study of a university CubeSat project and 18 semi-structured interviews. This is important to improve university management of funds and resources for applied research projects, which are becoming increasingly "projectified" and require heuristics for managing the different aspects of complexity. The findings show agreement on **lack of clear guidelines** for processes or methods for **managing projects** at the university. Moreover, there are different opinions on what constitutes good research management, and the results indicate that an agile approach and collaboration between technical and project management is positive for research projects.

University CubeSat projects face challenges related to high turnover, knowledge management, balance between coursework and project work, as found in the literature review (see Section 3.2) and reported in Papers A, C, D, and F [12, 14, 15, 17]. This can be addressed by employing agile principles [1] to providing an integrated approach to systems engineering and project management. The need for new tools, such as MBSE, for development of complex systems in university CubeSat organizations are addressed by Papers B, D, E, and F [13, 15–17]. This is done through an analysis of the HYPSO project team and the ways agile practices, such as Scrum, were able to improve the balance between coursework and project work, and contribute to mission success. University CubeSat missions are moving from being mainly educational to having more scientific purposes [6, 72, 85], and are participating in larger networks of collaborating systems, described in Papers G and H [18, 19, 35]. Finally, the increasing projectification of applied research in academia [48] is discussed in Papers I and J [20, 21], and how this pertains to Systems Engineering in two Norwegian research organizations.

Second, this thesis contributes to the existing literature on management of CubeSat projects in universities, confirming the known sociotechnical challenges. Methods such as the Agile Decision-Guidance Method, systemigrams, N2-diagrams, and technology readiness levels are shown to be helpful to understand projects and characterize the organization. Moreover, this thesis contributes by suggesting a set of factors that affect the projects to varying degrees in different phases: external facilities, internal facilities, parts supply chain, team knowledge, internal communication, and clear objectives. Other researchers in the domain could potentially apply these factors to improve their understanding of university CubeSat projects, and study the performance. Furthermore, this thesis contributes with more knowledge on how MBSE can be used in support of system design, especially in the an SoS context.

Third, the main practical contribution from this work to *reducing the time between mission concept definition and launch while still ensuring mission success* is the identification and implementation of an agile development approach based on Scrum [86]. This methodology is based on the findings from the longitudinal case study of the HYPSO organization and supported by a digital engineering [87] infrastructure. The analysis of the organization using the Agile Decision-Guidance Method highlighted which factors the CubeSat project could benefit from an agile approach, and what agile capabilities already existed. The foundation of the agile management approach is the ability to respond to changing circumstances and new discoveries. Most students lack experience with planning their work and engaging in multidisciplinary teamwork. An agile approach that considers coursework and project work, and continuously adjust the planning and prioritization of tasks depending on the academic calendar and external factors (such as supply chain, external test facilities). This is managed by including the students and their professors as stakeholders to the system delivered by the project team using Scrum, such that thesis tasks are a part of the overall backlog and features. This approach encourages systems engineers and project managers to coordinate and integrate their responsibilities and react to the dynamic environment of academia. The central, shared, digital information system serves as a repository and dissemination point in the process that supports the system development lifecycle and facilitates that the information is maintained and known by designers in all lifecycle phases and disciplines.

A list of the contributions and how they relate to the research questions is given in Table 1.1.

Table 1.1: List of Papers and Contributions to Research Questions

Paper	Contribution	RQ
A	Description of challenges for HYPSO project team. Adapting the ECSS review format to suit context. Using N2 and systemigram to improve understanding of team dynamics and dependencies, and introducing concurrent working times as well as shared cloud drive to increase communication frequency and lower barriers between team members.	RQ-0
B	Incorporating dependability results (a Failure Mode, Effects, and Criticality Analysis) in an MBSE system model with traceability to components, and actions to increase communication of dependability to designers.	RQ-3
C	Categorization of factors influencing development time for the HYPSO project team. Application of subsystem and system readiness levels to support management and prioritization of tasks.	RQ-0, RQ-1
D	Case study of how the HYPSO project team fared under the COVID-19 lockdown and presentation of technical considerations and management considerations that should be addressed. Importance of agile management and flexibility in schedule is emphasized.	RQ-0, RQ-1, RQ-2
E	Presentation of an integrated Digital Engineering approach with tools and processes for University CubeSat teams. Adaptation of the agile approach to include hardware, AIT, software, and coursework tasks to increase transparency for project planning.	RQ-1, RQ-2
F	Analysis of HYPSO project from an agile point of view in the customer problem space, the solution space, and the product development space which uncovered strengths and weaknesses of the organization to adopt agile practices.	Applied research projects in academia, RQ-0, RQ-1, RQ-2
G	Using an SoS approach to identify capabilities needed, use cases, and assess constituent systems for monitoring coastal regions.	MASSIVE as SoS
H	Modeling an SoS with MBSE for high-level scenario descriptions and architecture considerations.	RQ-3
I	Presentation of academics' views on systems engineering and project management, to create a basis for departments to allocate resources, training researchers, etc.	Applied research projects in academia
J	Increase understanding of complexity for managing research projects in academia.	Applied research projects in academia

1.5 Outline of Thesis

This thesis consists of: Part I — the theoretical background and key findings, see Figure 1.8 for an overview, and Part II — the appended papers.

Part I:

- Chapter 1 provides the introduction and objectives for this thesis.
- Chapter 2 provides some background literature as a basis for the readers.
- Chapter 3 provides the literature review which was the basis for defining the research questions.
- Chapter 4 describes the research design.
- Chapter 5 provides a summary of the findings presented in the papers.
- Chapter 6 examines and evaluates the findings, discussing the contributions and limitations of the research.
- Chapter 7 concludes the research and suggestions for future work.

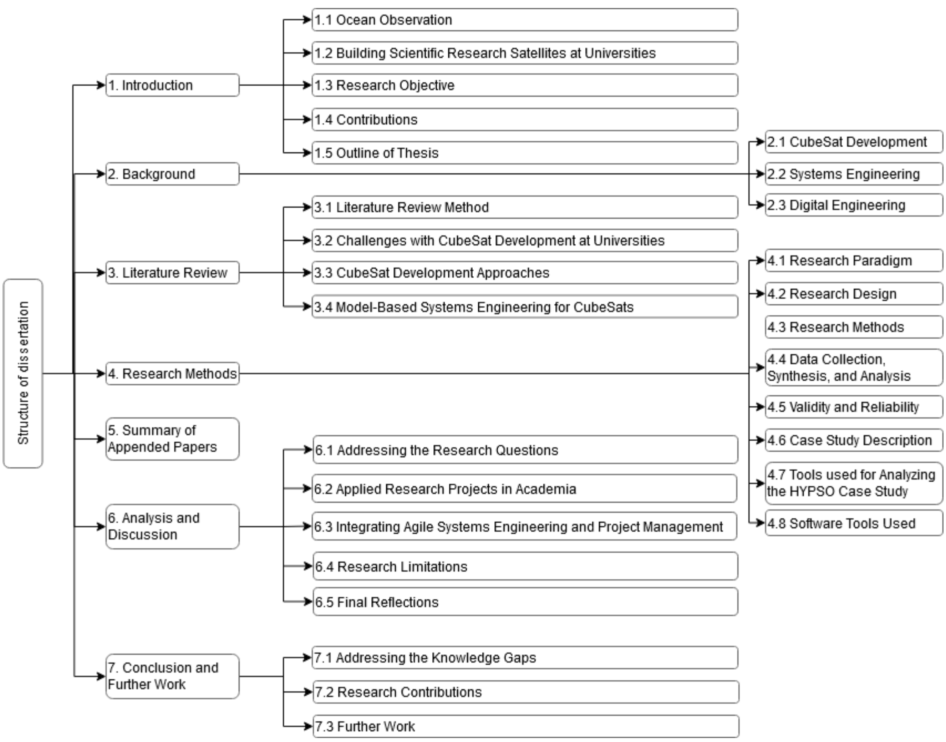


Figure 1.8: Overview of thesis structure.

Chapter 2

Background

The beginning of wisdom is to call things by their proper name.

Confucius

The following sections are intended to provide some background information about small satellite development and System-of-Systems, which are themes that will be referenced in the rest of the thesis and in the papers.

An overview of the chapter is given in Figure 2.1.

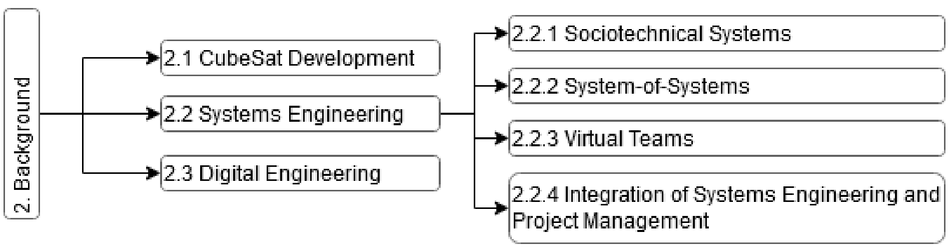


Figure 2.1: Overview of Chapter 2.

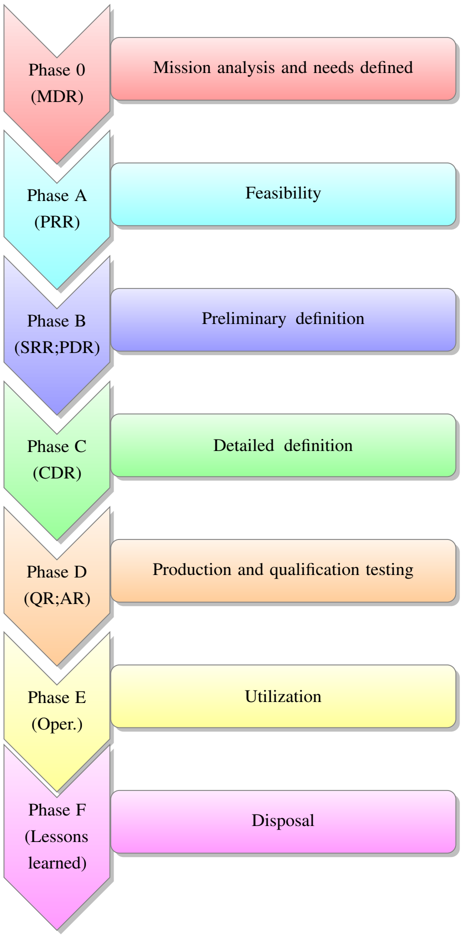
2.1 CubeSat Development

In Sweeting [6], the history and notable trends of small satellite development are recounted. Sweeting attributes the proliferation of CubeSats and small satellites to increased availability of highly reliable COTS microelectronics, better ground segments providing nearly global data link services, and increased "leftover" launch capacities. Small satellites are now used for technology demonstration, Earth observation, communication purposes, educational purposes, and deep space missions. In tandem with the growth of CubeSats in universities, the spread of the "NewSpace" environment has contributed to new funding mechanisms and development processes when compared to the "traditional" space projects. The NewSpace environment includes a larger ecosystem for private industry and businesses in space, offering remote sensing and communication services traditionally provided by governmental companies and initiatives [6].

This thesis focuses on a specific category of small satellites, the Cube satellites, or "CubeSats". If a CubeSat weighs less than $10\mathrm{kg}$, it is classified as a "nanosatellite." The concept was conceived in the late 1990s by Professors Puig-Suari and Twiggs [88]. Over the years, the CubeSats have grown from university educational toys into carriers for versatile scientific instruments and businesses [6]. CubeSats now perform various missions, such as communication systems relay [89] and earth observation [90]. There is a growing CubeSat community internationally, see Figure 3.2, where businesses, academic, and research institutions drive the technology onwards [6]. Projects can procure hardware, software, and services from a multitude of providers.

A typical lifecycle for small satellite development is shown in Figure 2.2. The phases may have different names depending on the standard and culture, and while it is shown as a top-down approach, it is often iterative. The phases can be adapted into other system development lifecycle models, such as the spiral model [91].

A top-down approach is usually recommended [92–94], starting with mission analysis and the establishment of mission needs. The mission analysis can include architecture considerations, but architecture with a CubeSat is usually chosen for small satellites because of the availability of COTS components [6]. From the mission analysis and needs, requirements are elicited and allocated to hardware and software subsystems. There is an increased effort to promote more agile approaches for, e.g., requirements engineering [80, 95, 96]. Depending on the cost and schedule of the mission, projects may choose to reduce performance or functional requirements depending on the availability of hardware and software subsystems.



- **Phase 0** — In Phase 0, it is essential to determine the mission objectives and needs. This needs to be communicated to the design team, the funding providers, and regulatory bodies. The primary objective of the HYPSO mission is to *"provide and support oceanography mapping through HSI and on-demand autonomous communications in a concert of robotic agents at higher latitudes in the vicinity of the Arctic/Norwegian coast."*
- **Phase A** — From the mission objectives, some mission requirements are identified to determine the feasibility of the mission and possible system designs. Typically, system designers are heavily involved in this phase to determine if and what is possible. Orbit analysis is performed, and preliminary budgets as input to system design trade-offs are established. For example, determining what data format was interesting to oceanographers, deciding how many images could be downloaded per day, and required ground stations.
- **Phase B** — During the preliminary design definition, the baseline design of the spacecraft is decided and verified through prototype testing and analysis. Throughout Phase A and B, the mission may also change due to technical limitations discovered. For example, prototype testing of the camera showed that it had to double its size to get enough light — and thus double the size of the spacecraft.
- **Phase C** — In Phase C, the baseline design is verified and ready for production. The final analysis is performed, and system budgets are determined.
- **Phase D** — For the last ground-based phase, the spacecraft will be integrated and verified for launch. The HSI spacecraft will be assembled at the satellite bus provider's premises, so the interface requirements and manual of the payload camera are the most critical inputs to this phase.
- **Phase E** — From the primary objective defined in Phase 0, it is clear that there are several actors and stakeholders just in operation. Ensuring that these stakeholders are considered during design is essential, for, after launch, it is too late.
- **Phase F** — For CubeSats, this is the most straightforward phase since they are typically launched into Low-Earth Orbit (LEO). After 5 years, the spacecraft will re-enter the atmosphere and disintegrate on its way down to Earth — so no disposal is necessary.

Figure 2.2: Satellite design process. MDR = Mission Design Review, PRR = Preliminary Requirements Review, SRR = System Requirements Review, PDR = Preliminary Design Review, CDR = Critical Design Review, QR = Qualification Review, AR = Acceptance Review. Adapted from the ECSS standards.

2.2 Systems Engineering

Systems Engineering (SE) centers on understanding the needs of the stakeholders and the context of the problem, before determining how to meet those needs with a system or product throughout its useful life. SE emerged as a discipline during the *Apollo* program when it became clear that the current working practices were not adequate to manage the unprecedented challenges of putting a man on the moon and returning him safely [97]. The discipline and practices have since evolved and continue to evolve, but the essence remains the same. In SE, it is critical to see both the parts (a reductionist approach) and how the parts come together to make a whole (constructionist view) and understand the problem in its context [98].

The ISO standard, ISO/IEC/IEEE 15288:2015, is the latest iteration of an actively maintained standard that elaborates the underlying processes that typically make up a system lifecycle [8]. This thesis adopts the INCOSE definition of systems engineering because it is widely recognized and includes relevant keywords such as stakeholder needs, requirements, verification, and validation [10]:

> *A transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods.*

The Venn diagram in Figure 2.3 shows the scope of SE and related activities, with the boundaries and overlaps with *System Implementation* and *Project Management*. The SE and PM roles should coordinate the activities shown in Figure 2.3, and organizations will differ in how they allocate the activities to multiple or single roles. Even though the systems engineers may perform their work as expected, this does not mean that the organization will deliver successful systems (in the eyes of the stakeholders). The result is dependent on all the other activities and parts of the organization also functioning well. However, the SE activities may help in achieving successful system delivery through their focus on collaboration and leadership [99]. The work in this thesis was mainly concerned with the left side of the Venn diagram, emphasizing the importance of considering people in the organization and the systems of interest.

The SE process begins with identifying the system's needs based on a stakeholder analysis or a gap analysis. Stakeholder analysis is an integral part of the SE process, and it is critical to ensure that the system delivers its capabilities according to stakeholder expectations throughout the lifecycle. The stakeholders can, for example, include funding bodies, the engineers doing the work, the people responsible for maintaining and operating the system, and the end-users of the service or capability the system delivers. Based on the stakeholder analysis, a concept of operation is developed, to establish a common understanding of how the system shall be used. Traditionally, the process then produces requirements and specifications that the system should meet, both in terms of function and performance. Different architectures and concepts are developed in parallel and even prototyped to see if the system design meets the specified needs. Concurrent prototyping and design allow for a

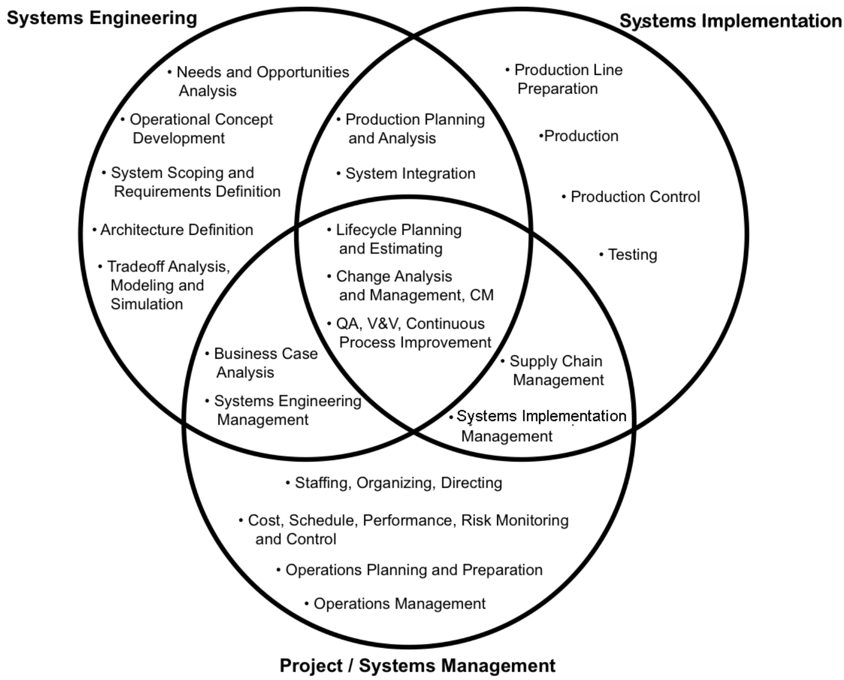


Figure 2.3: Venn diagram showing the system boundaries of Systems Engineering, Systems Implementation, and Project/Systems Management. SEBoK Original [100].

more circular and iterative SE process, and the stakeholders can validate and verify that the system meets the required needs continuously under development. Depending on the system's complexity, these prototypes may be anything from 3D-printed parts, basic user interfaces, drawings on a whiteboard, or basic delivery of services. As the understanding of the problem and the system design matures, the project may deliver high-fidelity prototypes to the stakeholders. Depending on the customer, SE artifacts such as interface documentation, test plans and reports, verification control documents, and requirements documents are delivered at maturity gateways to check and document the design in a stepwise and consistent manner.

Winter and Checkland [101] suggest the need for two viewpoints, *hard* and *soft*, each providing a contrasting image of managing projects. According to Crawford and Pollack [102], there is some confusion between PM's hard and soft paradigms. The hard paradigm is philosophically grounded in positivism and realism, while the soft paradigm is grounded in interpretivism and the constructivist epistemology [102]. Furthermore, the hard paradigm is often associated with a linear approach to problem solving and managing the project development lifecycle. In contrast, in the soft paradigm for project management, the manager is continuously observing and evaluating the situation and choosing to improve or change the project [101]. The hard paradigm can also take an iterative approach to problem-solving, but the goals and the path to achieving the goals are known and planned. Hard systems thinking aspires to "an efficient means to achieve a predefined and agreed end [103]." In contrast, soft systems thinking methods are based on "interactive and participatory approaches to assist groups of diverse participants alleviating a complex, problematic situation of common interest [103]."

One may surmise that a combination of both a *hard* and *soft* systems approach substantially benefits the project [104]. A well-known soft systems approach is the Soft Systems Methodology (SSM). SSM is concerned with "issues related to culture, value systems, attitudes, human perception, meaning, and learning in human activities, both at organizational and individual levels [105, p. 111]." These aspects are especially relevant for product development, where people do most of the creative work. Winter [106] provides a real-world example of the use of SSM to describe "a messy, complex situation [106, p. 802]" to improve a new project at Tesco, a UK-based grocery store chain. In this example, the SSM allowed for a more informed discussion among the stakeholders to take *purposeful action* to improve the project.

Some have claimed that SE employs "common sense" principles [98, 104, 107]. There have been efforts [108], most notably by Honour [109, 110] and Boehm, Valerdi, and Honour [91], to measure the benefits of SE activities in projects. Honour [109] found an optimal SE activity level in a project of $15 - 20\%$ of the total effort. However, this was a limited study with self-reporting primarily by systems engineers and their perceptions. Honour continued the research in [110] where technical quality and program success concerning SE activities were discussed and the results were consistent with the earlier study.

Furthermore, a division of effort to the different SE activities of *mission definition*, *requirements engineering*, *system architecting*, *system implementation*, *technical analysis*, *technical management*, *scope management*, and *verification and validation* was suggested. *Verification and validation* came out as the prime benefactors of SE activities [110]. Boehm, Valerdi, and Honour [91] looked at software projects and measured the Return on Investment of applying SE. They found a relationship between SE activities and software productivity and that even minimal SE efforts would increase the project productivity significantly [91]. Cook and Wilson [104] describe which types of activities yield the most significant value in a project's lifecycle and how the advent of MBSE may bring additional value to projects and be linked to traditional engineering activities. They also touch on what is necessary to have a *good* SE environment, such as clear and shared objectives, a common model of system and worldview, an understanding of the process, and a stable environment and context in which the system is developed and deployed [104]. Even so, there are obstacles and barriers to introducing SE in any organization. Some organizations believe that using SE *processes* may hinder creativity because they associate process with a prescriptive, detailed, flow-diagram approach that constrains your work process [111]. Sheard, Lykins, and Armstrong [111] continue to list other barriers to adoption of SE: poor definition or understanding of SE, applying SE without a specific purpose, and lack of resources.

2.2.1 **Sociotechnical Systems**

To develop complex systems, we need people, processes, and supporting systems. This trilogy defines a *sociotechnical* system or organization. A sociotechnical system is a system that contains the subsystems *people* and *processes*, and the *methods*, *facilities*, and *equipment*, as shown in Figure 2.4. The social and technical systems dynamically interact and evolve together [112]. Organizational systems that include humans are complex and messy [105], and we cannot analyze them in the same way as physical systems [101, 113, 114]. We need to understand the nature of sociotechnical systems better to improve our decisions for design and operations of the technical systems, processes, supporting systems, and the management of people.

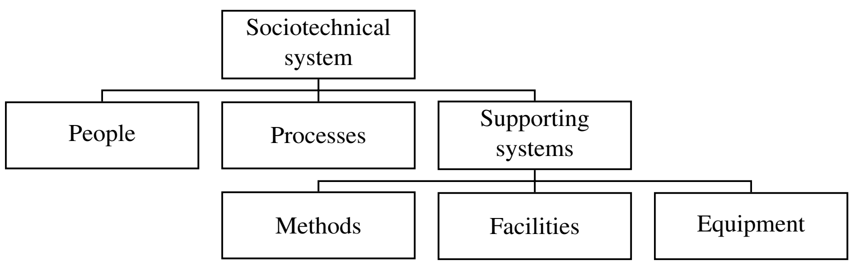


Figure 2.4: The sociotechnical system adapted from Pajarek [113].

A joint workshop between the US Department of Defense (DoD), NASA, and representatives from the aerospace industry need more research to be conducted on topics that include MBSE adoption and approaches, requirements engineering, modeling, sociotechnical issues, and education [115]. The "history and origins" of sociotechnical systems are outlined in Rouse et al. [116] together with the contemporary views. The authors describe methods and tools for sociotechnical systems, the complexity of sociotechnical systems, model risks, and future needs for understanding sociotechnical systems. As pointed out in Vaughan [117], the *Challenger* explosion was linked to organizational issues and decision-making [118]. In McDermott et al. [119], the sociotechnical dimensions for urban microgrids are modeled using the systemigram, described in Section 4.7.2 as one of the tools useful for describing the context. Heydari et al. [112] used agent-based modeling for the design and simulation of sociotechnical systems, which is especially appropriate as systems become more decentralized and include autonomous assets. The sociotechnical systems exhibit network structures (social media being a popular example) [116].

2.2.2 System-of-Systems

Applying a System-of-Systems viewpoint when designing systems has gained popularity as more and more systems are recognized as consisting of collaborative projects with different operational and managerial entities. Especially for solving the grand challenges, or national/international collaborations, or applying autonomous systems, the System-of-Systems (SoS) methods provide critical insights for decision-making and can support system development in different stages of the SoS lifecycle. An SoS is recognized as being an arrangement of standalone/independent Constituent Systems (CS) with *operational* and *managerial independence* [2, 9, 74]. The arrangement results in capabilities and allows for the fulfillment of objectives that no single system could achieve on its own. The SoS can be a permanent or temporary assemblage of CS. Characteristics used to describe SoS, in addition to the aforementioned are: *geographical distribution*, *evolutionary development*, and *emergent behavior* [9, 75, 120].

The critical characteristics considered for this research are (1) operational and (2) managerial independence [9]. A given CS can operate independently if the SoS is disassembled and they are acquired and integrated individually but contribute meaningfully to the SoS. The managerial independence accounts for the CS developed in different periods and with little or no attention to interoperability with the other systems of the SoS. Furthermore, the SoS may be classified into different types. The typology is continuous, and the SoS may be categorized as a combination of the types and change classification depending on the CS changes. The types are [75]:

1. **Virtual SoS** — where there is no joint management or objective, the CS may change depending on the mission, resulting in possibly emergent desirable or undesirable behavior and no clear maintainability strategy. For example the stock market.

2. **Collaborative SoS** — there are central players that make decisions for the SoS, but may not have the authority to influence the CS directly, the CS collaborate in various levels to achieve common goals. For example the internet.
3. **Acknowledged SoS** — there is central management, and the SoS has specific objectives to fulfill, the ownership and management of each of the CS may be different while supporting the SoS, any changes are managed through collaboration between the central management and the CS. For example city zoning.
4. **Directed SoS** — centrally managed SoS designed to achieve specific objectives, CS may still make decisions independently but the nominal mode is to achieve the central goal. For example a missile defense system or the deep space network.

In Dahmann et al. [121], the wave model is suggested as a guiding framework for developing SoS. The framework supports the SoS development by highlighting artifacts that SoS designers should focus on, such as establishing "requirements space" at a high level to support decisions regarding which systems to include and exclude, developing architecture alternatives, and integrating master schedules. Furthermore, the framework highlights the importance of managing the complex relationships between different operational entities through agreements and planning future updates of SoS. The iterative nature of the wave model in Figure 2.5 highlights this importance.

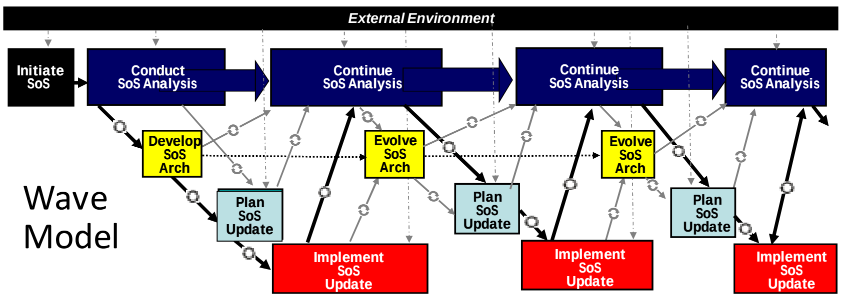


Figure 2.5: System-of-systems wave model from [121].

2.2.3 Virtual Teams

Virtual team management became a topic for this thesis work after the COVID-19 pandemic outbreak, which resulted in nationwide measures such as home office for all researchers, lockdown of the campus, and little or no access to lab facilities. Virtual teams may also be called "distributed" or "dispersed" teams, and the focus of a virtual team, as used here, is when the management and communication are conducted with virtual tools [11]. Managing virtual teams, also when planned for, comes with its own set of challenges. Virtual team management is enabled at a higher level by having good organizational support [122–124]. The sociotechnical viewpoint is also relevant for managing virtual teams. There are technical aspects such as training people in using digital collaboration tools and ensuring that the tools used enable high-quality communication [122]. The social elements include emphasizing continuous contact and communication, understanding the diversity of the team members, and expressing flexibility and empathy [11, 125]. Especially in a university setting, where the students live away from their families, team members may feel an increased sense of isolation compared to a setting where they see people in the lab or classes [11]. Furthermore, there must be processes for the virtual teams to establish expectations, such as planning, interpersonal, communication, and collaboration processes [124].

2.2.4 Integration of Systems Engineering and Project Management

This section will give a brief overview of the integration of SE and PM in complex applied research projects [126]. Some of this text is from Paper I [20]. In general, systems developed in the recent decades are more complex and require a higher level of coordinated engineering and management [126–128]. Systems Engineering (SE) and Project Management (PM) offer heuristics for improving the management of complex projects. The systems engineer "lead(s) the technical efforts necessary to developing the system [129, p. 1]", and the project manager for meeting the project targets. Both are responsible for ensuring that the system meets the needs of the stakeholders. They may have overlapping tasks and responsibilities, such as technical project planning, risk management, stakeholder management, and decision-making [129, 130]. To deliver a successful system, it is important that the systems engineer and project manager coordinate and integrate their approach, and that the shared responsibilities do not result in unproductive tension.

INCOSE, the Project Management Institute (PMI), and the Massachusetts Institute of Technology (MIT) established an alliance team in 2011 to analyze the integration of SE and PM, based on the recognition that these roles have overlapping and complementary responsibilities [131], see Figure 2.6. INCOSE produces a SE Handbook based on ISO 15288 that includes processes and guidelines for managing a system lifecycle. Concurrently, the PMI publishes the PM Body of Knowledge (PMBOK) that provides a similar guide for PM. Moreover, Oehman et al. [132] provide a guide for lean enablers and suggests methods for improving the integration between SE and PM [128].

Lilburn's [134] study recognized that through their shared concern of meeting a customer's needs, SE and PM should be aligned and coordinated through functional decomposition and by practical integration. In Browning [135], a highly-cited paper with an abstract including the keywords "systems engineering", "project management", and "integration", the use of Design Structure Matrix (DSM) to facilitate the modeling of system architecture, team and organization, activities and schedule, and low-level relationships is described. The interrelationships between system, team, activities and schedules could be explored using the DSMs, and thus provide more information to systems engineers and project management. Roe [136] describes the topic of integrating SE and PM in an Integrated Product Development setting. Smith and van Gaasbeek [137] use the analogy of the DNA double helix to represent an inherent need for integration of the two for project success. Johnson [138] discusses the history of and similarities and differences between

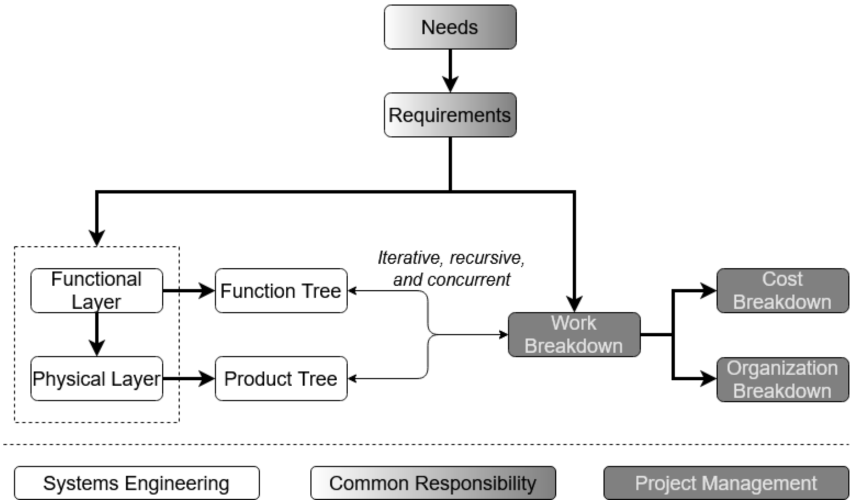


Figure 2.6: Project Breakdown structures and responsibilities between SE and PM. Figure from INCOSE PM-SE Integration Working Group [133].

SE, PM, and Operations Research. Other research suggests that sharing a common language and understanding responsibilities is necessary to make the integration work [130, 139, 140]. Barker and Verma [141] found that "application of formal SE with effective project management and test processes, can significantly improve development and integration productivity."

Xue et al. [128, 142–145] have published several papers regarding the integration of SE and PM. The research group suggest methods and tools for supporting the collaboration between SE and PM [128, 142]. A process for monitoring projects based on SE and PM project performance indicators is suggested in [144] and [143], and demonstrated on a university case study and a manufacturing company, respectively.

A recent study by Kordova, Katz, and Frank [129] identified the main processes shared between SE and PM by studying the SE Handbook and the PMBOK, performing an interview study of experts, and discussing what it meant for meeting the project targets and issued recommendations for successful integration. They found that up-front coordination and familiarity with each of the two roles are important for joint management, and documented in the project management plans. Furthermore, the two roles should agree on processes for resolving conflicts, use common tools, and agree on project goals. Finally, that the personalities of the individuals are a human factor that may enhance or detract from the probability of success.

In Jaafari et al. [146], the use of an information system that integrates the different discipline data is conceptualized. While not specifically describing how to integrate SE and PM, it describes some of the complexity with PM and need for information from different viewpoints. Sauser et al. [126] discuss the rise of SoS as highly complex integrated systems which have characteristics such as distributed governance and independent objectives. The authors discuss different paradoxes such as boundary, control, and the team (as a system), recognizing the need for integration of SE and PM to develop complex systems. Traditional PM has been criticized for restricting innovation and creativity [147, 148] because of its strict processes requiring detailed planning and scoping before starting the project. For innovation and product development, it may not be possible to plan a project at the level of detail required by these guidelines and restrict the solution space alternatives. The terms *research* and *innovation* may be used interchangeably in product development, but the terms may have different implications depending on how the end goals are framed or expressed. The *agile* project management approach is grounded in enabling the ability to respond to changing circumstances and discoveries. It also means empowering the whole project organization to participate in making decisions instead of relying on the project manager to decide the scope and team activities. There will still be project management activities, but the top-down hierarchical chain-of-command is replaced, reducing some of the asymmetries between power and influence. Simultaneously, agile project management is not equal to the absence of management [149], but rather a shared team leadership and management [147].

It is rare to receive training in both SE and PM, as they are typically separated in academic or training environments. Davidovitch et al. [150] apply a training simulator for training teams of project managers. Building on this, Cohen, Iluz, and Shtub [151] show how a training simulator for systems engineers can be used to teach PM. Cohen et al. concluded that simulations help gain the practical understanding needed [134] and that other scenarios should be developed for training and compared with real-life situations [151]. Furthermore, training in PM relevant to research projects is lacking [49], and in Xue et al. [145], the research gap of missing pragmatic integration is highlighted. Xue et al. [145] suggest a framework for aligning the processes, and demonstrate how the different standards and guides are overlapping and/or complementary. They conclude that although it may require a large effort, the changing mindset of how SE and PM should and could collaborate is critical to success.

2.3 Digital Engineering

[This text is from Paper E [16], and included as background information on Digital Engineering, which is referred to in this thesis.]

The digital transformation that is taking place in all elements of society calls for continuously updated knowledge for leaders and for engineers. The increasing project complexity introduced by the advent of embedded systems and Cyber-Physical Systems (CPS), and the tools needed for developing them challenges managers to re-think the approach to leading projects and people to ensure knowledge management and project success [152]. While this is challenging in industrial settings with experienced engineers and support systems, developing complex systems in an academic environment adds factors such as high turnover, coursework, lack of multidisciplinary teamwork experience, and fewer competent SE and PM resources. The Digital Engineering

(DE) definition of the U.S. Department of Defense (DoD) has been adopted for the work in this thesis: "an integrated digital approach that uses authoritative sources of system data and models and a continuum across disciplines to support lifecycle activities from concept through disposal" [87, p. 340].

Digital engineering goes beyond using computer tools to aid engineering, but includes the engineering process and approach to development. Choosing a DE strategy should be done based on the resources available and needs of the organization. A framework that assesses the DE competence was developed by the Systems Engineering Research Center (SERC) which looked at the following areas: *adoption*, *velocity/agility*, *knowledge transfer*, *user interface*, and *quality* [153]. While the framework did not specify how to measure the competence in each of the areas, it listed different factors and examples of processes or outcome metrics that could be used. Some factors identified can be categorized as objectives for why DE measures are incorporated, others as factors which may influence the adoption, and other factors as outcomes and direct competencies the organization can gain with DE practices. DE has a strong relationship with MBSE and Model-Centric Engineering (MCE), and establishing a "single source of truth" for a project [87]. However, there is currently no single solution for the whole system lifecycle to provide an authoritative source of truth. Most work-forces and organizations need to transition their methods and methodologies to DE and incorporate it into their engineering practices, and ensure possibilities for collaboration and information sharing throughout the system lifecycle between developers and the stakeholders. Most university CubeSat teams use some degree of DE, such as employing version-controlled software repositories, using CAD tools, shared cloud documentation, and using cloud-based issue tracking or project management tools to achieve integration in the management of knowledge [51].

Chapter 3

Literature Review

Each one of you can change the world, for you are made of star stuff, and you are connected to the universe.

Vera Rubin

A literature review which summarizes the challenges university CubeSat teams face is presented in Section 3.2, and introduces some of the research gaps this thesis attempts addressing. Section 3.3 provides an overview of different development approaches applied in university projects, and their advantages and disadvantages. An overview of the chapter is given in Figure 3.1.

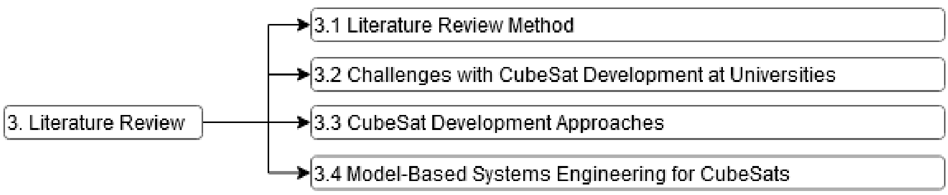


Figure 3.1: Overview of Chapter 3.

3.1 Literature Review Method

The primary focus of the literature review is to look for similar applications (e.g., CubeSats in universities, virtual student teams) and identifying the central issues mentioned in their findings. The literature review is meant to be representative, and focuses on the following journals and conferences: *Acta Astronautica*, *IEEE Systems Journal*, *International Journal of Project Management*, *Systems Engineering*, *INCOSE INSIGHT Magazine*, *Aerospace*, *IEEE Aerospace Conference*, *Engineering Management Journal*, *Small Satellite Conference*, *Conference on Systems Engineering Research*, *IEEE System of Systems Conference*, *INCOSE International Symposium*.

Google Scholar and Oria (the university library) are used for literature searches. The following search terms are used when collecting data for literature review for this thesis: *university CubeSats*, *project management CubeSats*, *system-of-systems cubesat*, *rapid systems engineering*, *agile CubeSat*, *agile hardware development*, *mbse cubesat* and *problems CubeSats*. This is a partial list as searches are done sporadically throughout the research. When the topics received many hits, the searches were narrowed down to only search for the keywords in the abstracts. Then, the abstracts were reviewed before reading the full articles.

3.2 Challenges with CubeSat Development at Universities

Small satellite development at universities has been focused primarily on CubeSat development since the release of the CubeSat design specification in the late 1990s [6, 154, 155]. The objectives for developing a CubeSat at universities have ranged from in-orbit demonstration of new technology, education and knowledge-building, the introduction of space to non-space faring nations, scientific exploration purposes, and communication purposes [6, 28, 51]. Known challenges for developing CubeSats at universities include (i) knowledge management in student-driven projects because of students graduating yearly; (ii) incorporating lessons learned systematically; (iii) schedule overruns and lack of funding; (iv) little system-level testing; (v) lack of formal methods for risk and failure analysis; (vi) successful integration of CubeSat engineering into the curriculum [23, 25, 26, 28, 51, 83, 95, 156–158].

Although there have been a large number of small satellites launched by universities, and still more are in the pipeline, as shown in Figure 3.2, 30-40% are dead-on-arrival in orbit or die within the first three months [23, 25]. Attempts have been made to understand why there is still a high percentage of failed missions, (Cf. [23], [25], [51], and [24].) In addition to unknown faults, in [23], the analysis of in-flight failures show a high percentage of Electrical Power Subsystem (EPS) faults at 0, 30, and 90 days in orbit. Another strong contributor is the communication subsystem after 30 days in orbit. In [24], failures are counted before launch. The data from [24] shows that the structural subsystem is the main failure driver, and electronics-related design faults the second most common failure driver. A common recommendation is to increase the time spent on testing at the system level; performing end-to-end testing with a ground segment and day-in-a-life of the satellite; ensure some knowledge transfer and lessons learned; limit the development time to 2–3 years, and apply a minimum level of processes and standards when developing the system.

Langer et al. [160] and Faure et al. [24] suggest two reliability estimation methods to find the "recommended" time needed for functional testing, and demonstrate estimation methods on their respective CubeSats. However, at their level of maturity these are not straightforward to apply without having a close to finished design in terms of selecting components and interfaces, which are needed as inputs. It is also not clear how much work is entailed to get a total picture when considering the complete CubeSat and all its subsystems. In Latachi et al. [158], an approach using Failure Mode, Effects, and Criticality Analysis (FMECA) and Failure Detection, Isolation, and

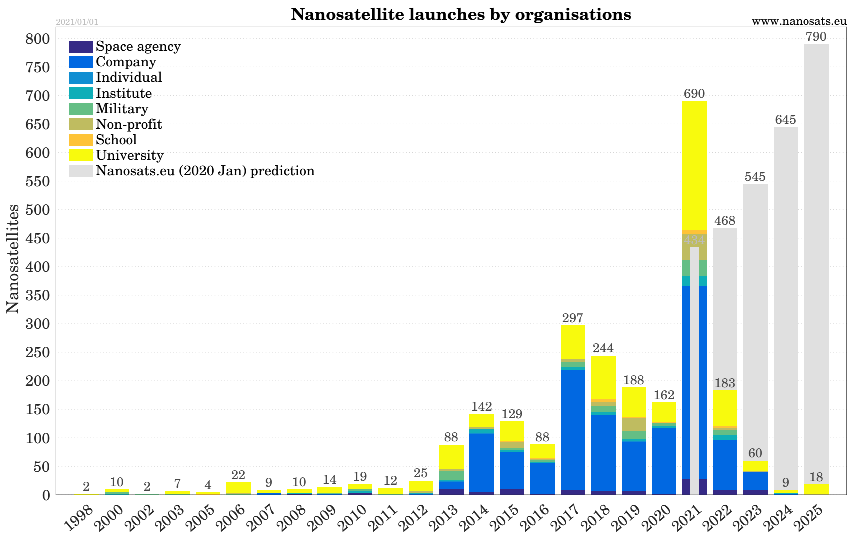


Figure 3.2: Number of small satellites launched per year, categorized by institution type [159]. The author of the figure uses the term nanosatellite, which denotes satellites weighing up to 10 kg.

Recovery (FDIR) for improving the reliability of CubeSats is presented, however this was at the qualitative level since COTS reliability data is not available. They found that the approach is helpful in defining corrective measures to mitigate or prevent the risks. Similarly, Menchinelli et al. [161] use FMECA for their CubeSat and subsystems, and introduce a nomenclature for easy reference of failure modes. However, FMECA analyses are subjective, especially when applied to a system composed of COTS components. It is also critical to include these analyses early in the design process, and integrate the reliability work as part of the whole design work, which sometimes happens as an afterthought. This can enable the developers to make decisions early to improve the design.

In Cho and Mazui [22] and Berthoud et al. [51], recommendations for team composition are also introduced, such as having 2–4 PhD candidates, 10–15 students, and 1–2 passionate faculty members supporting the teams. From the literature review, the development of CubeSats in universities can be considered a sociotechnical system, as stated in RQ-0, and should be approached as such. The challenges and identified research gaps are listed in Table 3.1, addressing the first research question, RQ-1, of this thesis.

Table 3.1: Summary of key challenges for university CubeSat teams and identified research gaps.

Key findings	References	Research gap
Spending time for up-front planning and then continuously updating plans is important to anticipate workload and tasks. The planning effort is often underestimated by university teams.	Berthoud and Schenk (2016) Langer et al. (2015); Swartout (2013) Alminde and Larsen (2009)	There is a lack of awareness of the value of front-loading for projects and tailored methods for project planning for university CubeSat projects are needed.
On-ground functional system testing is key to increasing mission success. Many failures could have been discovered by performing more end-to-end or test-as-you-fly testing of the integrated CubeSat prior to launch.	Zaidi et al. (2019) Berthoud et al. (2019) Alanazi and Straub (2019) Venturini (2017); Faure et al. (2017) Langer and Bouwmeester (2016) Berthoud and Schenk (2016) Corpino (2014); Swartout (2013) Alminde and Larsen (2009)	A pragmatic, conscious, integrated approach to testing in university CubeSat teams is missing. University CubeSat teams are either unaware of the need for tests or often lack resources to follow traditional standards for verification and validation. Better onboarding information is needed to provide the necessary instructions.
Knowledge management is challenged in university CubeSat teams by high turnover because of the academic year, and low possibility of predicting or controlling team composition in terms of background and experience.	Berthoud et al. (2019) Campos et al. (2020), Kleespies (2017) Berthoud and Schenk (2016) Langer et al. (2015); Segret et al (2014) Birkeland and Gutteberg (2013) Bouwmeester et al. (2008)	An approach for managing knowledge and the availability of an up-to-date information system is lacking. Methods for improving the project planning methods to adjust for resource shortages, and understanding what influences the development time are needed.
Maintaining a vision of the mission, requirements, and project scope is critical for motivation and establishing a shared understanding of the problem domain for university CubeSat projects. Dedicating time in the first phase of the project lifecycle with mission requirements is important.	Alanazi and Straub (2019) Venturini (2017) Larsen and Nielsen (2011) Dubos et al. (2010)	Additional research is needed for defining a pragmatic approach to work with vision, mission, requirements, and project scope in a team setting.
Having a risk management process can increase the probability of mission success. Most university CubeSat projects need a low-effort approach to risk management because of scarce resources.	Menchinelli et al. (2018) Venturini (2017); Faure et al. (2017) Langer and Bouwmeester (2016) Gamble and Lightsey (2014) Guo et al. (2014)	Additional practices are needed to define ways to integrate the reliability and risk management activities into the design process so that they are treated as a valuable design input and can be traced.

3.3 CubeSat Development Approaches

Waswa and Redkar [52] reviewed "non-traditional" approaches to space systems engineering, where one of the categories was "small satellites." They found that needs strongly influenced approaches for the whole sector in the small satellite development area. A key distinguishing factor between small satellite approaches and traditional approaches is the approach to handling risk. Risk-averse behavior coupled with a push to include many functions onto one satellite influences the traditional approaches, leading to long development times and large, monolithic spacecraft. The new development of approaches include increased reliability, shorter development cycles, and increased use of COTS components. A comparison of popular university CubeSat development approaches are given in Table 3.2.

Traditional stage-gate development processes, such as the waterfall method, are characterized by being linear, deterministic, risk-averse, often starting at the top of the planning and then breaking work packages down into tasks, and with customer involvement at set intervals [77]. A solely agile approach is appropriate when there is high task uncertainty, a focus on learning and early testing, self-managing teams and continuous customer involvement [77].

Using NASA and ESA Processes for Space Mission SE and PM

Both NASA and ESA have established processes for designing a space mission and have made efforts to streamline the activities for CubeSat developers through the CubeSat 101 publication [53] and the ECSS In-Orbit Demonstration (IOD) tailored standards for CubeSats [55]. The CubeSat 101 publication target audience are "CubeSat developers working with NASA CubeSat Launch Initiative (CSLI) [53, p. 2]", and is focused on the lifecycle and milestones related to the CLSI. This includes the support that CubeSat developers get through technical reviews from experts. However, it is helpful for providing an overview and best practices of how to develop a CubeSat. The ECSS CubeSat IOD standard provides a tailoring of applicable ECSS standards for use in ESA IOD missions, which have different funding mechanisms and stakeholders from most university projects. The tailoring provides insight into typical activities that CubeSat projects should perform, but requires an understanding of how the ECSS body of standards are structured and how to apply them to be of use. The ISO standard for lean satellites [54] outlines different testing strategies and levels of environmental testing recommended to qualify a "lean satellite". It does not provide a process description of how a satellite project should be, or what activities should be performed throughout the lifecycle. It is strongly recommended to use established standards for both the system's development and interfaces [22, 162, 163], by relying on established data buses and interfaces instead of inventing in-house designs. This can lower the cost of and shorten the time of development, and many COTS components adhere to established industry standards.

Databases for Component Reference Designs

Efforts have been made to share lessons learned and approaches to CubeSat development to increase the probability of a successful mission. For example, some authors recommend using a database of components to support in the conceptual design phase (Phases 0-A/B in Figure 2.2) [164–167]. Depending on built-in capabilities, the database takes orbit and spacecraft characteristics, mission needs, and other parameters as inputs and provide a set of components that the designers can choose from, or compares alternative architectures and provides performance and cost parameters. Drawbacks with these approaches are the maintenance of such databases and that the databases do not consider how designers might configure certain components after purchase.

Concurrent Engineering

Other sources suggest applying Concurrent Engineering principles [68, 93, 167–169], especially in the first phases of development. Concurrent Engineering is based on having a common work-space and iterative work sessions where disciplines feed into a shared model to maintain an updated system design. It enables rapid, iterative design cycles where developers are up-to-date on system budgets and the impact of their changes to the whole system. Loureiro et al. [57] present a Concurrent Engineering methodology which addresses more than just the first phase of development, which they have successfully applied for the past 20 years. ESA has a "Concurrent Design Facility" in which mission planning and conceptualization are regularly performed. The ESA facility also employs the principle of co-location, where all designers are present in the same room working in teams, and have a systematized approach to collaborative design cycles. A challenge with Concurrent Engineering is the exchange of information from computer models (such as mechanical and thermal analysis), which should feed into the system model. However, the iterative design cycles makes it an attractive approach for missions where not all is known beforehand and there is a need to have frequent reviews of design concepts.

Agile Practices for CubeSat Development

Agile practices help develop systems when we cannot know all the requirements or technologies to be used in advance and need to have an approach to planning which allows for adaptation based on new knowledge or needs from the customers. Similar to concurrent engineering, agile is based on iterative design cycles with frequent feedback from stakeholders of the product. The agile workflow attempts to reduce the "cone of uncertainty"/"funnel curve" [170] by increasing the design iterations and dealing with change and disruptions early. Using agile practices and methods for SE has gained popularity in recent years [58, 77–79, 147]. It is defined as "the ability to respond effectively and with competence, to operational environments with increasing uncertainty and unpredictability [171, p. 861]." Today, the agile concept is often associated with software development and Scrum [86, 172, 173], a method to apply agility in the organization.

The agile concept has also influenced modern SE. At its core, agile SE focuses on continuous learning and modification of processes and project goals [79]. Some principles for agile SE that have been suggested include (1) focus on delivering customer value, (2) team ownership, (3) embrace change, (4) continuous integration, (5) test-driven, and (6) taking a scientific approach to systems' thinking [149, 174]. The Scrum method describes a team structure (a product owner, developers, and a Scrum master); a workflow with sprints lasting 1–4 weeks; a sprint planning meeting where each sprint's tasks are agreed upon and allocated to the team; a daily Scrum (often called a stand-up) that acts as a short status meeting; a sprint review, in which a Minimum Viable Product (MVP) (an agreed upon iteration of the system) is demonstrated and reviewed by the product owner, and tasks are assessed if they conform with the "definition-of-done"; and sprint retrospective where the team analyzes how the sprint went and if there are possibilities for improvement [86]. In Bott and Mesmer [173], a literature review of agile SE based on Scrum is presented, and a suggestion for how Model-Based Systems Engineering (MBSE) is suitable for agile SE (MBSE is explained in the following section). Bott and Mesmer highlight how a Scrum approach is suitable when there are "uncertainties surrounding the system development [p. 84]," and suggest that more validation is needed to see if the framework can be generalized. Darrin and Deveraux [175] compare agile and design thinking and explore advantages and disadvantages with the different methodologies, and what they can add to SE processes to meet the uncertainty challenges of using COTS components in systems, or responding to changing requirements.

The application of agile practices in space industry was discussed in Carpenter and Dagnino [176], as well as the authors suggesting areas for how agile approaches could support safety-critical analysis. Since then, more work has been published about using agile practices in the space industry, but most of this section focuses on agile for university CubeSat projects. ESA published a handbook for "Agile software development" in 2020 [80], based on the agile manifesto [1] and the Scrum methodology [177]. The purpose of the handbook is to be a reference for ESA projects and provides guidelines for (i) when to select an agile approach; (ii) what kind of agile approach to select; and (iii) a reference model for how to apply agile to software development.

There have been reported benefits from applying agile practices to software development of CubeSat systems [95], which enabled early system-level testing as recommended by [51], or mission development approaches [178] (i.e. early-stage concept exploration). In Könnola et al. [179], agile practices are shown to help communication and knowledge sharing in the Finnish space sector. Campos et al. [78, 180] apply a tailored agile/lean approach, and combine the role of SE and PM, recognizing the overlapping activities and responsibilities, for their ManitobaSat-1 project. Agile practices have also been applied to CubeSat missions in Garzaniti et al. [83]. They highlighted that the agile approach helped tackle unforeseen changes, although there is still work to be done on how to involve testing activities of hardware in a Scrum approach. A key difference between software and hardware Scrum is the specialization of team members, for where in software there is more overlap between people, in hardware the team members are usually more specialized [181]. This makes it more challenging to share a sprint and tasks. In Böhmer et al. [182], a Scrum approach to managing the hardware development of an exoskeleton is presented. Böhmer et al. suggest grouping tasks related to hardware development, similar to the “user stories” in Scrum for software development, and recommend an agile approach for projects where there is incremental development and high usage of prototypes. Hardware Scrum is challenging because of lead times associated with manufacturing or the reliance on subsuppliers. This can be mitigated to a certain degree by using 3D-printing or other rapid prototyping techniques, or by choosing another definition for MVP. Mosher et al. [183] provide a pragmatic approach to how hardware Scrum can be performed, and make suggestions for deliverables as the end of each sprint. In Lee et al. [181], agile guiding principles for hardware are suggested:

- *Incomplete, fabricable prototypes over fully featured models.*
- *Collaborative, flexible teams over rigid silos.*
- *Improvement of tools and generators over improvement of the instance.*
- *Response to change over following a plan.*

However, how to execute this practically with a university CubeSat team should be further explored. The agile practices can be associated with a steep learning curve, with new nomenclature and routines that need to be taught. Additionally, teams require some time to “learn” task estimation. While there seems to be a growing adoption of agile practices in university CubeSat projects, more research on how to use agile practices for hardware, software, and Assembly, Integration, and Test (AIT) are needed and should be shared [176, 179].

From Table 3.2, using Concurrent Engineering or an agile approach seems appropriate for university CubeSat projects which are characterized by high turnover, unknown requirements, and an unstable academic schedule — and both offer transferable skills development, which is an asset for the students and may be an added motivation factor. There is little evidence of how well agile approaches work with AIT and hardware CubeSat systems, or how it may be used by project managers to manage the schedule better. This leads to the elicitation of the second research question, RQ-2.

Table 3.2: Advantages and disadvantages of different development approaches.

Agile approaches	
Agile approach for software development and/or hardware development.	Campos et al. (2020/2021) Garzaniti (2019) Mosher et al. (2018) Selva et al. (2016) Berthoud et al. (2015) Segret et al. (2014)
Disadvantages of agile	
Not many "successful" sprints in the traditional sense of Scrum, but overall progress as scheduled. Adjustment time for new teams to understand Scrum methodology. Requires an integrated product owner's participation.	
Advantages of agile	
Responsive to external and internal changes. Enables self-organizing teams and can create greater team member ownership. Transferable skills development for future career employment.	
Concurrent engineering	
Concurrent engineering for concept phase development of university CubeSats. Based on having a common work-space and iterative work sessions where disciplines feed into a common model to always have an updated system design.	Pereira et al. (2019) Ehresmann et al. (2019) Menshenin et al. (2019) Arnaut et al. (2013) McInnes et al. (2001)
Disadvantages of CE	
Setting up and managing the integrated system model across multiple platforms. Agreeing on common work sessions across coursework.	
Advantages of CE	
Integrating engineering disciplines to complete design cycles faster. Transferable skills for future employment.	
Database approaches	
Using databases for component selection and architecture development to support trade-off studies.	Jacobs and Selva (2015) Triana et al. (2015) Chang et al. (2007) McInnes et al. (2001)
Disadvantages of databases	
Maintaining the database (inclusion of new components, collecting and recording information from datasheets). Setting up the database and calculation of relevant parameters for concept development when not all providers provide the same level of information. Does not account for configurability without engineering analysis.	
Advantages of databases	
Aids rapid selection of components/orbits/architecture. Contributes to understanding of dependencies between subsystems. Ensures a common knowledge base for all team members.	

3.4 Model-Based Systems Engineering for CubeSats

Model-Based Systems Engineering (MBSE) has been hailed as a paradigm shift in SE, changing the reliance on project development processes toward model-based artifacts. One of the main drivers for this shift has been to increase the shared understanding of a problem or a design by removing some of the misconceptions that different interpretations of natural language can introduce [62]. The cognitive barrier to documentation is not as high as models. The possibility to use MBSE rather than solely document-based SE for design is through a shared ontology and semantics. The visual aspects of MBSE allow for modeling the system from different viewpoints and sharing the information between viewpoints without having to ensure that all documentation is concurrently up to date. As the systems become more complex and the organizational aspect more complicated, the shift is strengthened in lockstep with increased digital capabilities and interoperabilities.

An early mention of MBSE and discussions of challenges and opportunities can be found in the 1998 INCOSE INSIGHT Magazine: *Model-Based Systems Engineering: A New Paradigm*. For example, the first article by Lykins and Cohen [184] list changes and advantages as follows:

- *Integration of modeling techniques across multiple disciplines and perspectives*
- *Evaluation of new and innovative modeling techniques and use of existing techniques for new purposes*
- *Exploring ways in which the systems engineering process should change as the use of computer-interrogable models replaces textual documents*
- *Development of a taxonomy to organize product modeling techniques and to identify the interfaces that would ensure semantically rich exchange of information between models, especially across technical disciplines*
- *Development of a taxonomy to organize process modeling techniques and to identify where different modeling techniques must be integrated.*

Fast forward ten years to 2009 INCOSE INSIGHT Magazine: *Model-Based Systems Engineering: **The** New Paradigm* [185]. While one could think that a decade would have given time to solve some of the challenges, it is clear that there is still no clear consensus on what MBSE is or how it should be introduced into an organization. The importance of modeling and moving away from solely document-based systems engineering is still emphasized, and different industries have done this in different ways according to their unique situations. Cloutier highlighted in the introduction to the magazine issue: "...that the specific tool, or language, or approach, is not the important thing; rather, systems engineers should model to understand the problem and to communicate with others about the problem. If your modeling approach helps you accomplish that, it is a good thing [63, p. 7]."

The literature review included application of MBSE to many of the system lifecycle aspects. There are many examples of successful MBSE use in concept and architecture definition and system design (see [186–189, 189, 190]), in requirements management (see for example [65, 191–193]), reliability studies (see [194–197]), and integration of verification and validation activities in MBSE have also received attention, (see [67, 198–200].) Agile and MBSE have been discussed before, (see [201, 202]), grounded in how many MBSE environments are software-based and lends themselves to agile practices. MBSE tools have existed since the early 1990s [203], but only lately have the tools matured sufficiently to provide critical interoperability [64, 204]. Some are waiting for the "super-tool" which integrates and enables interoperability of all disciplines [201]. However, while many view MBSE as the solution (see for example a study by Huldt and Stenius [204] where 50-75% of respondents reported improvement of SE activities by using MBSE), it is still unclear if using MBSE provides the value it promises [205].

The INCOSE Space Systems Working Group (SSWG) has developed a CubeSat System Reference Model (CSRM) as a platform-independent and SysML-compliant model [70]. The purpose of the model is to enable both universities, government entities, and industries to develop CubeSats systematically. The development work of the model started in 2012 and was submitted to the Object Management Group (OMG) in 2020. It is the first model-based specification submitted to OMG. Several papers have been published about the development of the model [67, 187, 189, 206–212], and recently about applying the model to different projects [69, 188, 213, 214].

The model is a logical representation of a CubeSat system, including stakeholders, requirements, behaviors, architectures, and technical measures. The CSRM is a non-populated logical representation, meaning that there are established relationships and traceability between the elements, but that the specifications of a given satellite are not given. The model also includes "help-texts" in the form of notes on each landing page and levels, which can help teams populate it for their specific mission. A reusable model is of great interest to programs such as MASSIVE, where it is expected that the spacecraft bus will not change significantly between missions, and neither will the mission requirements nor the objectives themselves. By reusing a model, system designers could update the components and behaviors, or add new requirements and stakeholders, without redoing the entire spacecraft and mission design [201],

While Cloutier's statement above ("...that the specific tool, or language, or approach, is not the important thing; rather, systems engineers should model to understand the problem and to communicate with others about the problem. If your modeling approach helps you accomplish that, it is a good thing [63, p. 7].") was true, and still is, today's distributed team structures and highly complex systems have an uncompromisable requirement for more interoperability to enable communication about the problem than what was available in 2009. The emergence of highly autonomous systems and larger SoS combined with a highly sociotechnical context, drives the need for more robust interoperability of the modeling tools and approaches [75, 115, 215, 216].

Even with the successes in applying MBSE, it requires training of engineers and organization commitment to deliver benefits. Voirin et al. [202] suggest the following factors necessary to adopt MBSE practices: that engineering teams see the growing complexity of systems and need new solutions to manage them, a skilled and motivated team of people with different backgrounds, shared will and objective to solve engineering issues, a responsive agile tool team, strong sponsorship and commitment from management, investment in skilled support to deployment, and a growing community of users sharing experiments. Similar findings were reported in [204, 217, 218]. Moreover, Huldt and Stenius [204] highlighted a research gap on "how to introduce and manage MBSE initiatives" and that there is a "lack of knowledge to integrate a model-based approach with current business processes".

The motivation of applying MBSE for the HYPSO project was two-fold: (1) The opportunities for increased communication across disciplines, reliability traceability, and model reuse for multiple satellites; and (2) opportunity for SoS modeling in the larger MASSIVE context. This resulted in RQ-3.

Chapter 4

Research Methods

Systems thinkers mentally model systems and parts of systems to simplify and understand structure and behavior. These models are fluid and constantly updated, and often support the ability to communicate complex systemic nature in simpler, more approachable terms.

Ross D. Arnold and Jon P. Wade (2017)

This chapter describes the choices of the methods and tools applied during this research. While each of the individual papers includes brief descriptions of the respective methods used, this chapter outlines the overall approach to the research design and includes reflections on the challenges and choices made through the process, which influenced the research design and its evolution.

A researcher's knowledge and personal experiences affect the research process, the formulation of research questions and data collection, and the conclusions made. In general, we cannot observe events neutrally or objectively, and our backgrounds include biases that influence what we see and how we see it. Though untraditional in the discipline of electronics engineering, my home institute, I chose the main research paradigm to be qualitative research, with the support of quantitative methods where applicable.

The research process starts with an idea or the discovery of a problem or gap in prior research, which inspires the need to investigate further to discover something novel or reduce the effects of the problem. This results in a research objective. The researcher plans to conduct the research based on the research objectives, including the development of research questions or hypotheses, starts collecting data, and then analyzes the data to answer the question or prove or disprove the hypothesis. This process is non-linear, consisting of many iterations where the researcher moves back and forth between the phases. This thesis is the result of the final synthesis of the analysis over many iterations.

The sub-sections in this chapter describe the research design and its evolution, and the research methods applied, the data collection and the analysis approach, see Figure 4.1. I detail the synthesis of data and comment on the reliability and validity of the research. Section 4.7 presents the methods used in the HYPSO project case study.

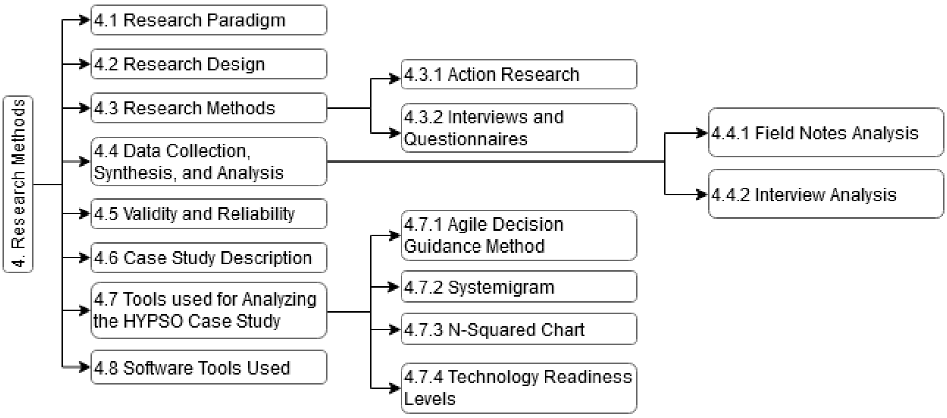


Figure 4.1: Overview of Chapter 4.

4.1 Research Paradigm

A paradigm is a term used to describe how we view the world and determine our perceptions of truth, and in a scientific community this means a shared way of thinking to solve problems within that field [219, 220]. In the "hard" sciences such as the natural sciences, Kuhn argues that scientists must have a shared set of exemplars to push the research forward, to solve the puzzles or to reexamine accepted theories together. A paradigm includes ontology (what we believe to exist), epistemology (what are the sources of the knowledge and how do we know it is true), and axiology (our ethics and what we believe is truth) [220]. Together, our paradigm shapes the methodology (our approach to systematic inquiry) applied to how we study the world.

In a trans-disciplinary research design, there will be different paradigms to consider when choosing the methods. Specifically, research methods used in SE and PM must be considered, as this work bridges and integrates the two fields. Szajnfarber and Gralla [118] describe the qualitative methods for engineering and SE research, and offer the following process steps for research design:

(i) **Choosing if a qualitative approach is needed:** The qualitative approach is recommended when the phenomena are poorly understood or when assessing new tools, methods, or approaches.

(ii) **Choosing the research focus:** Typically guided by research *questions* instead of stating a hypothesis.

(iii) **Select cases:** Things to consider when selecting cases, including how generalizable the case would be, variability between cases, and how much in-depth knowledge can be gained from the case.

(iv) **Data collection:** Using well-established and known techniques for data collection, scoping the data amount, ensure validity and reliability.

(v) **Analyzing data:** Similar to the previous step, ensuring that the data analysis approach is appropriate to the data set and questions under consideration — and ensure validity and reliability of results.

(vi) **Describing the results:** Qualitative research often results in descriptions of how and why systems do what they do or how they are used. These results can be used for decision-making and future design improvements of systems.

(vii) **Writing theory:** Unlike other disciplines where qualitative research approaches are applied, SE papers are often shorter, and it will be challenging to write enough theory to illuminate the results.

Within systems engineering, we look to *systems thinking* for philosophical underpinnings. We use a *reductionist* viewpoint (breaking into constituent parts, often viewed as "hard" methods [221]) to understand each constituent element. A good systems thinker also needs the *constructionist* viewpoint (looking at how an element interacts with other constituent systems to produce a behavior) to understand a system in its context. This combination gives us the skills as systems thinkers to "consider both the forest and the trees [222, p. 11]."

The nature of this research requires a dual approach between the hard paradigms (realism, positivism, quantitative methods) and the soft paradigms (idealism, interpretivism, qualitative methods) because it deals with both humans and systems in a sociotechnical research setting. This thesis takes a social constructivist stance, recognizing that the world is independent of human minds but that any knowledge we build is based on human construction.

4.2 Research Design

The research design has evolved continuously during the PhD period. The research objectives came out of the MASSIVE and HYPSO project needs, as explained in Section 1.1.1. These research objectives focused the initial literature review and stakeholder analysis which resulted in the knowledge gaps and motivations for the research questions outlined in Chapter 3.

The overall research process is shown in Figure 4.2, starting with selecting a research area and the overall objectives. These are grounded in the research group context I joined (the MASSIVE project). Literature review is performed continuously, and at the start used to identify research gaps and develop research questions. To answer the research questions, a data collection method is planned, and then performed. Based on continuous data analysis, the research questions are refined, and the results are incrementally published. Finally, this thesis is produced as a synthesis of the research process and overall contributions to addressing the final research questions.

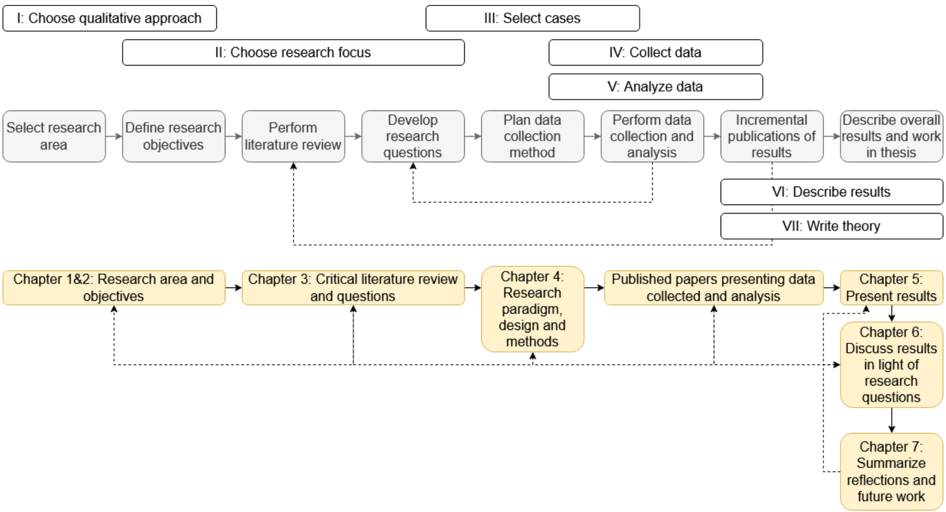


Figure 4.2: Overall research process. The white boxes are the steps described in [118], overlaid with the research process steps in grey to show the flow and when each step was most active. The yellow boxes show which parts of the thesis reflect results from each step. Solid arrows signify the plan, and dashed arrows the feedback of reflection and iteration.

The research questions start out broad, more akin to research objectives, and they are improved upon through iterations with colleagues and supervisors as more is learned about the research area and from the results.

Different data collection and analysis methods are chosen to account for validity and reliability in addressing the research questions. The results for each of the research questions are discussed in the papers as indicated in Table 1.1, and each publication has its associated theory. An overview of the relationship between the knowledge gaps, the research questions, and the data sources is given in Figure 4.3.

COVID-19 Pandemic The COVID-19 pandemic had a significant impact on this research and gave opportunities to explore new managerial challenges. The overall schedule of the satellite was affected when the campus locked down, and team members were not allowed to access the lab or facilities. The HYPSO project also found supplier management more challenging, such as access to external test houses or surface treatment facilities, and import of COTS components from abroad. For project management, there was a shift to fully digital management of the team. New processes had to be introduced, and a realization of how the risk management approach had failed to fully consider the consequences. The analysis of the effects the lockdown had on the HYPSO project can be found in [15, 16], with a more in-depth review of relevant literature for the topics.

Table 4.1: Research Design based on [118].

ID	Process step from [118]	Considerations for this research
i	Choosing if a qualitative approach is needed	The challenges the MASSIVE project faced cannot be extracted from the context and the organization, and in order to assess if improvements can be made, a qualitative approach was needed. There was existing literature from similar teams, but understanding the overall empirical context was important for improving the HYPSO organization.
ii	Choosing the research focus	The overall goal was broad — improving the success rate of university-led CubeSat projects, which acted as a guiding principle for the elicitation of more precise research questions. These were continuously iterated upon.
iii	Selecting cases	While multiple cases would have enhanced the data and analysis, the research focused on a longitudinal replication case study following a CubeSat project from its inception to launch. The single in-depth case study gives sufficient information to evaluate the research questions. The semi-structured interview study was to improve the understanding of the university context and SE and PM.
iv	Scoping and conducting data collection	Different data types were used: existing literature, interviews, questionnaires, observation notes; sampled at intervals when there were events or milestones for the CubeSat project in addition to continuous observation.
v	Analyzing data	Looking at key concepts such as design reviews, trade-offs, requirements management, project planning. Finding patterns or challenges within these key concepts, comparing to other literature or theory.
vi	The "result" of qualitative research	Description of how agile practices and a Digital Engineering workflow can support applied research projects in academia (drivers of performance affecting development time, description of context). Description of how academics view SE and PM, which may influence management of applied research projects.
vii	Writing theory	A mix of systemigrams, narrative descriptions, and diagrams to describe and analyze what has been found in the study.

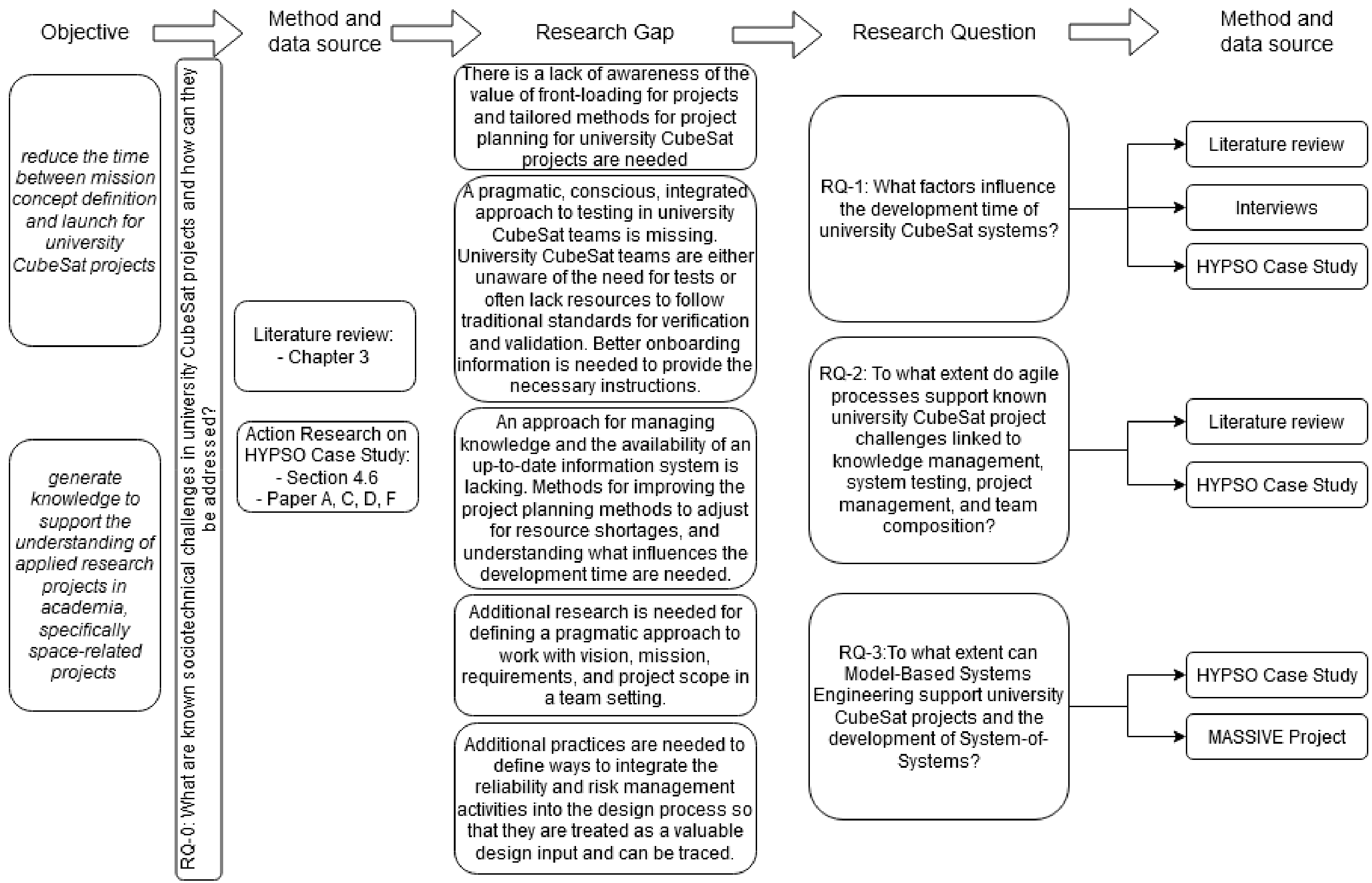


Figure 4.3: Overall research approach relating knowledge gaps, research questions, data sources and research methods. While this is shown as a linear process, there has been continuous feedback and refinement of research questions, knowledge gaps, and methods.

4.3 Research Methods

Caillaud et al. [223] describe the state-of-the-art of research methods for SE. They compare research in SE to both research in general engineering and to research in information systems. Their findings show a low level of validation of research in engineering and recommend best practices for validating results in SE research [223]. They recommend using *case studies* and *grounded theory* approaches for research in SE, which use methods similar to *Action Research*, shown in Figure 4.4. Their similarities lie in the iterative and cyclic approach to research. However, one of the main differences between grounded theory and Action Research is the amount of researcher first-person participation.

4.3.1 Action Research

Action Research is applied in many fields such as health care, education, and social studies [224, 225]. Action Research is based on the iterative process shown in Figure 4.4 with repeating cycles [226]. The objective of this research is to generate knowledge to support the understanding of applied research projects in academia, specifically projects related to space systems. This knowledge can be applied to create a process with supporting methods to *reduce the time between mission concept definition and launch while still ensuring project success* through the application of SE and PM tools and practices. Action Research is concerned with improving practice continuously by evaluating the system, planning, acting, and describing the state. Furthermore, since I was participating in the HYPSO project in a number of leadership roles, it was difficult not to take action to improve the practice. The Action Research described in this thesis is categorized as *participatory* action research [227]. The "natural and intuitive" approach is concurrently the main criticism of Action Research, where some may find it lacking in rigor and hard to distinguish from practical work [228].

The research project in this thesis is about changing and improving the process – not "(. . .) solely to understand social arrangements [229, p. 93]." The strength of the method is that the practitioner is a part of the research, engaging directly with the users and the processes. Through continuous collaboration and communication, the practitioner can both identify and introduce new processes and methods into the research and evaluate them as they are being employed. The emphasis is on collaboration between the researchers and the users. In my PhD research, I chose this method instead of the *silent observer* role where the researcher is "outside" the research environment. The changes can be practical and valuable to the HYPSO organization and not get lost as some PhD candidate's single-person effort [229], but only by active participation and clear valuation of this collaboration.

Application of Action Research in the HYPSO Project Multiple themes were investigated with Action Research within the HYPSO project team to see which processes worked well and why. The main cycles and actions are highlighted in Figure 4.4. The resulting papers from these actions

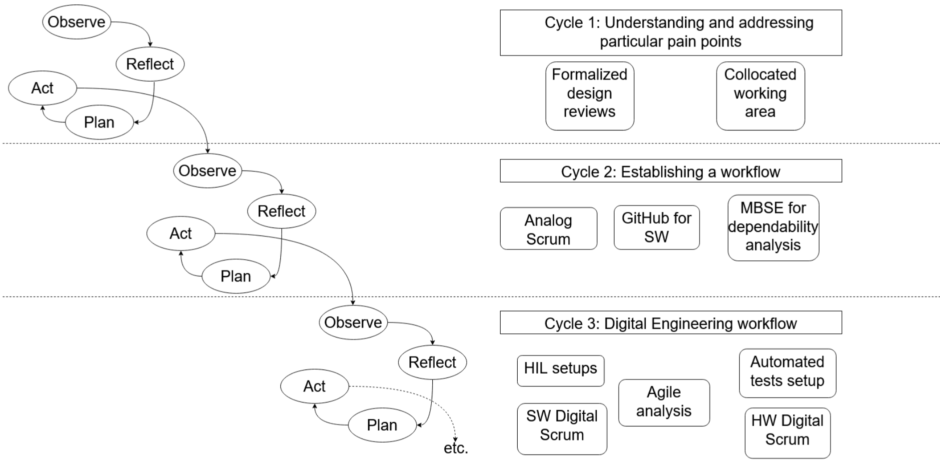


Figure 4.4: Action Research model with the cycles applied during the research period.

are shown in Figure 6.2. There is a cyclic nature of learning-as-doing. This means observing and reflecting on the team, drawing from personal industrial experience and literature, and testing actions continuously as the project manager and researcher, and getting feedback from participants as well as observations to record what worked and what needed to be re-evaluated. The selection of themes is based on the literature review and initial analysis of the project as presented in [12]. The plan-act-describe-evaluate cycle shown in Figure 4.4 took place several times, and improvements or actions were planned to correlate with the project events (such as in the onboarding period, after a design review, etc.). Many of the evaluation events happened during the weekly management meetings in the team, and through lessons learned sessions. There are three cycles highlighted in the Figure 4.4, these are the macro-cycles. There were also micro-cycles, e.g., figuring out the best way to set up GitHub for software, or how to incorporate testing using Hardware-In-the-Loop (HIL) setups in the Digital Engineering workflow.

4.3.2 **Interviews and Questionnaires**

Interviews may support case studies by collecting data that are not available quantitatively [230]. Individual interviews can be conducted as structured interviews, where all respondents are asked the same questions; or, in semi-structured interviews where the interview guide directs the questioning; or by unstructured interviews where the interviewer has a theme in mind but no formal interview guide. The questions in an interview may be open-ended or simple yes/no questions, depending on what data the interviewer needs to collect. However, questions may be poorly phrased and introduce unwanted bias into the data, or reactions or body language from the interviewer may unintentionally bias on the interviewee [231], and inaccuracies in analyzing the data are more likely if the interview was not recorded. In Hove and Anda [230], a review of interviewing in engineering research is given, as well as recommendations, which have been followed for the interviews conducted in this research. Questionnaires may be used for collecting survey data and may suffer from some of the same biases as interviews.

Application of Interviews and Questionnaires For Paper C, the interviews were conducted as online video conferences and recorded. No interview guide was prepared, but the interviews focused on the themes discussed in the paper. Some informants were asked to describe interfaces and subsystems either by sketching or commenting on a sketch.

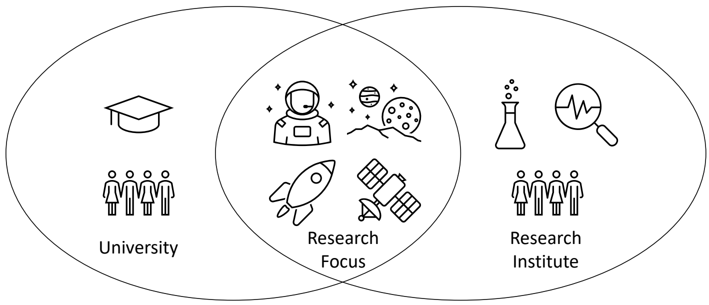


Figure 4.5: Representation of the participants for papers I and J. They both do research projects in the space domain. All subjects had a MSc degree, and some had PhD degrees. The organizations are in the same city.

Three surveys using questionnaires were conducted during the course of this thesis. One was concerning why students joined the HYPSO-1 project, and was to support the field notes analysis. The second was concerning the use of tools during the Critical Design Review (CDR), and to improve the design review, see Paper D. The third was for participants in a SE course, to gain an understanding of what they knew of SE before starting the course (Papers I and J). The interviews for papers [20] and [21] were conducted as semi-structured interviews of 18 participants, each between 45–60 minutes. See Figure 4.5 for an overview of interviewees background.

Sample: The participants were selected based on their involvement in space projects in two university-based institutions, using a key informant sampling method [232] as shown in Figure 4.5. They all have MSc degrees, and many have a PhD degree. The objective of the interviews was to explore how academics understand SE and PM. The key informant sampling method was chosen to collect in-depth information relevant to the research questions, with a mix of people representing different roles within their respective space-based projects. The informants perform research based on space knowledge, are also responsible for technology development, and practice applied research [48]. For paper [14], the interviews were conducted as online video conferences and recorded. No interview guide was prepared, but the interviews focused on the themes discussed in the paper, and some of the informants were asked to describe interfaces and subsystems either by sketching or by commenting on a sketch.

4.4 Data Collection, Synthesis, and Analysis

The data collection methods used were review of documentation, observation and field notes, physical artifacts, literature review, questionnaires, open-ended and semi-structured interviews. Using a variety of data collection methods in the qualitative study supported a broader range of interpretations and viewpoints. The data analysis software tool NVIVO [1] was used for literature review and interview coding. The interviews were coded according to the questions in Table 4.2 (Q-1-1, Q1-2, etc.) The tools listed and described in Section 4.7 were applied to analyze some of the data.

4.4.1 Field Notes Analysis

For the Action Research, notes were taken throughout the PhD research and were often reviewed during the data analysis. Notes were typically taken after an event related to the research questions happened or after interactions where requirements, mission, information sharing, design reviews, etc., occurred. When discussing these themes and research questions, the notes were reviewed to analyze patterns and refresh the initial conclusions drawn at the time.

Field notes analysis was conducted by reviewing field notes at intervals during the HYPSO project case study. These were not coded in NVIVO, but any reflections used and published were reviewed by other team members to validate the insights. The field notes were also used as a reference when discussing different issues such as design reviews or requirements within the HYPSO project team, in my role as a project manager, to collectively reflect on what could be improve in future project work. This was very useful for recollection of events.

4.4.2 Interview Analysis

The interview analysis was mainly centered on the 18 semi-structured interviews used as basis in Paper I and Paper J. Other semi-structured interviews were conducted as part of the HYPSO project case study. Most of these were recorded using a video call system and took place online (because of the COVID-19 pandemic lockdown). The analysis of these interviews was done by reviewing them and noting the important insights from the informants as relevant to the topic discussed. For example, for Paper C, the informants were asked to describe their subsystem, to explain their interfaces, and what were challenges for them when developing their subsystems. This information was used to identify which factors the team members identified as important when considering development time.

The 18 semi-structured interviews used in Papers I and J.
The following text and Table 4.2 are from Paper I, and included here for reference. Three researchers were involved in the data collection and analysis.

[1] A qualitative data analysis software https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home

The interview protocol was based on the research question developed through an iterative process using relevant literature. A semi-structured interview format was chosen for a natural flow of a dialogue and allowed the interviewer to ask additional questions, if needed [230]. Introductory questions such as “Tell me a little about your background.” or “What educational background do you have?” started each interview to help the informant relax and build rapport. The questions were posed in a combination of descriptive, (“How would you describe the research process?”) and reflexive, (“What would you say are the benefits and challenges of systems engineering?”) questions. All interviews were carried out face-to-face and recorded, and the interviewer took notes in case the recording was lost. A single researcher acted as the primary interviewer for all informants, while the second researcher listened and asked additional questions if needed at the end. The third researcher did not participate in the interviews, and only analyzed the transcriptions. The questions were available in two languages, and interviews were transcribed in their original language, either English or Norwegian. The informants were anonymized prior to analysis, and only the interviewer had the key to match the informant to a transcript.

An interview analysis protocol was based on Likert scales [233] of 1–5 (1 = to a low degree; 5 = to a high degree) with different statements the researcher would evaluate, given in Table 4.2. The interviews were analyzed independently by three researchers to provide triangulation on the results. The statements were based on the research questions, in addition to an evaluation of to which degree the informant had an engineering, educational, or research stance. The assessment of stance was based on how the informants identified themselves (for example if they said they were engineers, or if they said their primary role was as a lecturer), and what types of tasks they said they did in their jobs. The first round of analysis took place over the course of ten weeks, where the researchers independently evaluated the statements based on the interpretation of the transcripts. After that, the researchers met and discussed the results and explored the differences in rating where applicable. Finally, a score, S_x, was assigned to each statement based on the median of the researchers' scores. A median was chosen because it gives a measure of central tendency based on the rank of the score, appropriate for the non-continuous nature of Likert scales data. The Likert scale was further compacted to three levels: *low* for levels 1–2, *neutral* for level 3, and *positive* for levels 4–5 to enable more accessible discussion of results.

Spearman's ρ correlation coefficient [234] was used to measure the relationships between Likert scale scores assigned to the fourteen protocol categories. The equation is given in Eq. 4.1.

$$\rho = 1 - \frac{6\sum d^2}{n(n^2-1)} \tag{4.1}$$

where d is the difference between the ranks of the median Likert scores, and n is the number of questions. We also calculated the *p-value* for each ρ-value to signify statistical significance at the $\alpha = 0.05$ level. The coefficient measures the tendency for ranked values to change together. A so-called monotone relationship has a value between -1 and $+1$, where -1 is perfect negative monotonic, 0 is no monotone, and $+1$ is perfect positive monotonic. A perfect positive monotonic means that all data points in X increase as Y increases, and a perfect negative monotonic means X decreases as Y increases.

Table 4.2: Statements used to guide the researcher assessment of interviews. Q1-Q4 were evaluated on a Likert scale. Q5-Q9 were open-ended questions that were evaluated based on overall impression and direct quotes from the interview transcripts. The RQs given here are not the same as in the overall thesis, but rather, the ones used in Paper I.

ID	Topic	Relevant to RQ in [20] no.
Q1	*Understand the academic stance of the Informant*	
Q1-1	To what extent does the Informant hold a research stance?	
Q1-2	To what extent does the Informant hold an educational stance?	
Q1-3	To what extent does the Informant hold an engineering stance?	
Q1-4	To what extent does the Informant understand systems engineering?	RQ3
Q2	*Understand the stance/definition/explanations of project, process, task, and goals.* *Understand how the Informant balances between processes and goals*	
Q2-1	To what extent does the Informant distinguish between engineering project and research project?	RQ1,RQ2
Q2-2	To what extent does the Informant distinguish between the engineering process and research process?	RQ1,RQ2
Q2-3	To what extent does the Informant distinguish between engineering tasks and research tasks?	RQ1,RQ2
Q2-4	To what extent does the Informant distinguish between research goals and engineering goals?	RQ1,RQ2
Q3	*Understand if systems engineering could contribute towards research processes and goals*	
Q3-1	To what extent does the Informant believe that SE is integrated in academia?	RQ3
Q3-2	To what extent does the Informant believe that SE should be integrated in academia?	RQ3
Q3-3	To what extent does the Informant believe that SE could be integrated in academia?	RQ3
Q4	*Understand the stance on different types of management*	
Q4-1	To what extent does the Informant distinguish between research and engineering management?	RQ2
Q4-2	To what extent does the Informant distinguish between research and project management?	RQ2
Q4-3	To what extent does the Informant distinguish between project and engineering management?	RQ2
Open-answer questions for the analysis		
Q5	What are the greatest benefits of systems engineering?	RQ3
Q6	What are the most challenging aspects of systems engineering?	RQ3
Q7	What, if anything, separates an engineering project from a research project?	RQ1,RQ2
Q8	What, if anything, would be the benefits of more knowledge/support to project and engineering processes in academia?	RQ1
Q9	To what degree did the SE course influence the Informant? What thoughts does the Informant have about the course?	

4.5 Validity and Reliability

Four types of validity are typically addressed for case studies: *construct validity* (i.e., "identifying correct operational measures for the concepts being studied"), *internal validity* (i.e., that the inferences made are correct and all rival explanations have been considered), *external validity* (i.e., "showing whether and how a case study's findings can be generalized"), and *reliability* (i.e., "demonstrating that the operations of a study — such as its data collection procedures — can be repeated, with the same results") [231, p. 42]. Yin [231] suggests measures for addressing validity and reliability, and based on this, a summary of the measures applied are shown in Table 4.3.

For the case studies, multiple sources of evidence (meeting notes, interview notes, formal and informal project documentation) and critical informants have read the drafts and discussions before publication to address *construct validity*. A limitation to the data collection and *construct validity*, highlighted for replication case studies, is confirmation bias, defined as: "Confirmation bias, as the term is typically used in the psychological literature, connotes the seeking or interpreting of evidence in ways that are partial to existing beliefs, expectations, or a hypothesis in hand [235, p. 175]."

Table 4.3: Methods for Addressing Validity and Reliability

Validity test	**Phase of research**	**Method**	**Limitations**
Construct validity	Data collection	Multiple sources of evidence; multiple methods of inquiry; key informants reviewed the drafts of reports and papers.	Qualitative research and social settings are unique, construct validity is difficult to achieve. Confirmation of researcher's preconceived notions [231].
Internal validity	Data analysis	Address rival explanations; selection of case study that reflects the context studied; pattern matching.	Research bias in selection of case and phenomena studied; bias in coding of data. Incorrectly assuming causal relationships.
External validity	Research design	Validation by comparison to similar case studies and theory.	Choice of research objectives may influence what is looked for.
Reliability	Data collection	Field notes taken through case study; methods described in each paper.	Not possible to repeat a case study after it has been concluded. My interpretations are embedded in the field notes, researcher bias.

Since replication case studies either confirm or disconfirm claims, there is a risk that the researcher will look for evidence to support (or not support) claims, and disregard evidence that does not support the objective of the study. For *internal validity*, there is a possibility that inferences have been made without knowing the whole picture. The maturity and history are clear rival explanations for the primary case study. Critical informants reviewing and discussing drafts contributes to addressing this validity test. The *external validity*, i.e., the transferability of the interpretations, has been addressed by comparing with other theories and similar case studies. Finally, for addressing *reliability*, it is impossible to replicate any case study perfectly. However, the methods have been described, and researcher triangulation was done for the interpretations reported in Papers C, E, F, I, and J. In addition, the process for data analysis has been described as well as the procedures for collecting data. The advantages and limitations of data sources used are given in Table 4.4.

Field notes, when collected over time, can show the historical progression and suggest long-term trends in an organization, which are not apparent in day-to-day work. Questionnaires can be used to gather data quickly from large groups and to collect trends over time. Open-ended interviews facilitate the participants' reflections on the topics discussed without being guided by a specific formula or form that the questionnaire would give. The quality of the research in terms of reliability and validity is discussed in Section 6.4. Figure 4.6 builds on Figure 4.2 with the assessment for the different research questions.

Table 4.4: Advantages and Limitations of Data Sources Used

Data type	Advantages	Limitations
Documentation	Source of progress of project and evaluation of milestones/goals. Can be reviewed repeatedly.	Can be incomplete and biased (reporting bias), often written to e.g., get a good grade or positive impression to reviewers.
Observation	Enables direct observation of events, interdependencies, contextual understanding of actions.	Not possible to observe everything, observer bias/selectivity, bias due to my involvement in events. Not able to observe results not in evidence — e.g., a future launch and deployment of the satellite.
Physical artifacts	Can support evaluation of performance [118] and mission success.	A physical artifact does not tell the whole story. The artifact does not support prediction about future performance.
Questionnaires	Can provide large amounts of data in short time, useful as source of patterns. Can be analyzed quantitatively.	Asking good questions, that minimize researcher bias in how the questions are posed, and which questions are asked.
Interviews	Targeted interviews can give insights and explanations, and personal views of the interviewees. Support in-depth exploration of a topic.	Challenging to ask good questions, response bias, reflexivity (that the interviewee says what the interviewer wants to hear [231].

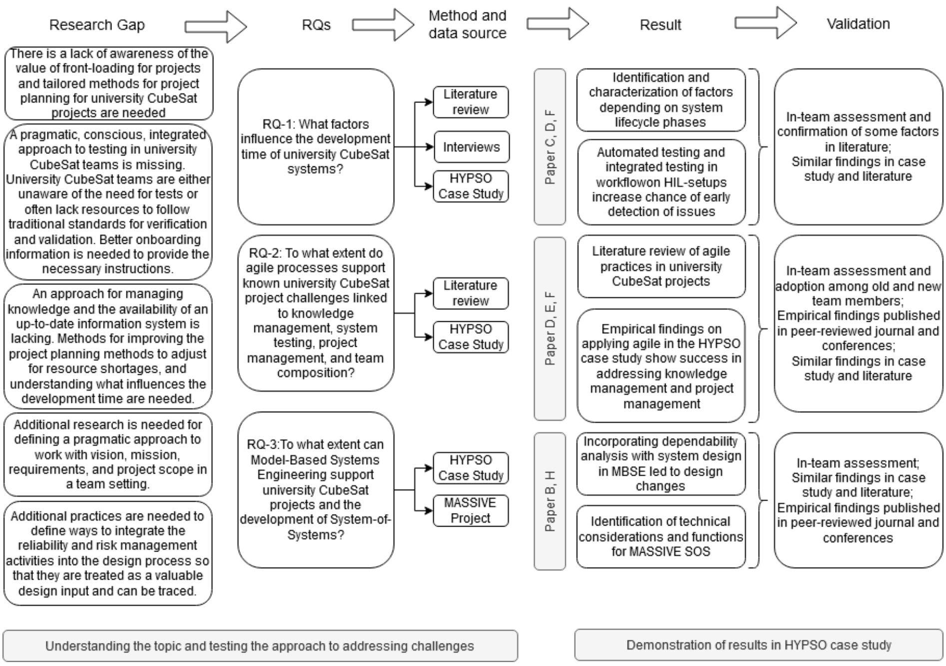


Figure 4.6: Summarizing the validation/assessment of the results discussed in Chapter 6. More details for each RQs validation can be found in the associated section.

4.6 **Case Study**

Replication Case Study The work in this thesis is based on a replication case study, which means to "assess whether a research finding from previous studies can be confirmed [236]." The findings from the literature review (see Tables 3.1 and 3.2) are the grounds for comparison. Since the basis of comparison is literature, a down-selection from the available articles was done based on the available information regarding the characteristics shown in Tables 4.5 and 4.6. In the literature I have searched for confirmations or disconfirmations, and explored these where relevant in the Results (Chapter 5) and Discussion (Chapter 6) chapters. A replication case study approach is suitable when building theoretical knowledge from empirical data, where each new case study can support the claims (or disconfirm claims) for the theoretical knowledge. The added value of a replication case study increases the more different the population studied is, to show that they are valid even for very different cases, or to attempt to disprove the claim [236]. However, all interactions between people are contextually situated, and the context cannot be replicated. This is an inherent limitation when comparing case studies and conducting replication case studies.

As a basis for comparison, a summary of the cases from the literature review and their population is given in Tables 4.5 and 4.6, built on [237]. In Table 4.5, the characteristics of the Aalborg University and Delft University CubeSat projects are given. It is not clear if they have adopted agile practices for their newer projects, but these cases provide a picture of successful CubeSat programs that have produced many successful missions. Table 4.6 presents the characteristics of two newer projects that have recently published results on their application of agile practices. Finally, Table 4.7 presents an itemized list of the characteristics of the HYPSO case study which is the basis of the replication case study. More details are given in the following sections. The comparison is limited to conference and journal articles that provide details on the case studied, and limited to university CubeSat projects. Therefore, there are variations in details describing each case, as this is dependent on the information available. Furthermore, there is a limitation in the self-reporting in articles, where the authors may have focused on certain aspects they find interesting, while other authors would focus on other topics. Some of the inferred findings are compared to the overall literature in Results (Chapter 5) and Discussion (Chapter 6).

The HYPSO project started its CubeSat activities in 2017, similar to the Iris CubeSat team, so these cases are concerning their first satellite. Aalborg and Delfi-C3 have multiple successful CubeSat projects under their belts since their start of activities in early 2000s, and it is not clear if the Unnamed CubeSat from [83] has any prior experience. The HYPSO team size is slightly larger than the Iris CubeSat and the Aalborg project, but smaller than the Delfi-C3. The team composition is similar across all teams, except that the HYPSO project also uses PhDs and Post Doctoral researchers actively in engineering activities as well. The HYPSO project resembles the Aalborg and Delfi-C3 projects that have thesis assignments for work and receive academic accreditation, while this was not clear from the Iris CubeSat or the Unnamed CubeSat. There is a mix of funding sources for all teams, and there was no evidence that the funding was an issue for the different projects. There are differences in amount of institutional and faculty support, where Aalborg is at the lower end of support together with the HYPSO project, while the Delfi-C3, the Iris CubeSat and the Unnamed CubeSat all receive institutional and engineering support from professionals. The workforce flowthrough of the HYPSO project team and the Aalborg team is similar, with student members changing each semester, but a core team remaining. For the Delfi-C3 CubeSat, the team changes each year. The other two projects have not described this. There is a large variance in the amount of in-house facilities, where the Iris CubeSat, the Unnamed CubeSat, and Delfi-C3 all have access to vibration table and thermal vacuum chamber. The HYPSO project and Aalborg projects do not, but have the basic workshops.

The cases have similar disciplines involved. However, Aalborg university has a space research program and Delft university (the host of Delfi-C3) have master programs in aerospace engineering and systems engineering. It is not described what study programs are available at the Unnamed CubeSat project. The HYPSO project does not have systems engineering programs. However, this has been addressed concurrently with the HYPSO-1 CubeSat development, where there has been hiring of an SE associate and adjunct professor, as well as new courses for students to take. The choice of COTS components and in-house development of subsystems varies between the projects. At Aalborg, almost all subsystems are in-house developed, grounded in the project based learning program. At Delfi-C3 it has been a mix of COTS approach and in-house development, similar to the Iris CubeSat, the Unnamed Cubesat, and the HYPSO CubeSat that both developed the payload in-house. For the HYPSO project, prior experience from student projects and industry influenced the decisions on when to use COTS components and when to develop in-house, as well as an understanding of resource needs and mission objectives. Developing systems in-house requires a lot of resources, and usually takes longer than one person's thesis work. While adapting COTS components for space [46] requires less effort. On the other hand, the resulting system could be less tailored for the specific purpose. In terms of management, the cases chose different approaches. For the Aalborg projects, students are encouraged to do the management themselves, gaining non-technical skills that will help them in their future work. The Delfi-C3 project on the other hand, has a dedicated project manager and adopted many standards for PM. The HYPSO project and the Iris CubeSat both lead the work using PhDs/faculty members. All teams adopt some type of formal design reviews. In addition, the teams adopting agile (Iris CubeSat, Unnamed CubeSat, and HYPSO CubeSat) also conduct the agile events such as stand-ups and keep a kanban board as well as version control of software. For documentation, the Iris CubeSat teams describes an approach with everything available on a team google site, and the Unnamed CubeSat does not describe it in detail. The HYPSO project uses a kanban board connected to the version-controlled software repository, and writes the documentation in the cloud so everyone can contribute. Futhermore, the HYPSO team has used MBSE for requirements management and traceability, which also acts as documentation for the design.

The Aalborg projects describe using PM practices mostly for scheduling, and have coordination meetings with systems engineers from each sub-group. The Aalborg project focuses its SE activities on requirements management and system budgets management. Similar to NTNU and the HYPSO project, most of the students do not have courses in SE or PM prior to joining the project. The Delfi-C3 project reports applying a structured approach to PM with artifacts such as work packages, phasing, scheduling, documentation management. Similarly for the SE approach, the Delfi-C3 team uses a structured approach following the standards, and produce much of the documentation typical in space projects. The Iris CubeSat project also takes a self-organizing team approach, and incorporate agile with a kanban board that is also connected to a gantt chart. The monitoring of the project is based on verification activities, and they use a cloud system for managing the requirements. The Iris CubeSat project appears to be an integrated approach to PM and SE with agile practices.

My prior background in Systems Engineering and Project Management has given me experience in different development approaches, both stage-gate, agile, and hybrid approaches. This influenced the choice of development approach for the HYPSO organization. After a stage-gate approach failed to provide the desired effects, with few people keeping to task deadlines, not seeing dependencies, and too much time spent updating a schedule that failed to consider academic coursework, an agile approach was chosen. Furthermore, I had experience with using the ESA ECSS standards from industry. While these are applicable when you have a large organization with multiple functions (and people to fill these functions with specific backgrounds), I quickly understood that this would not be possible in an academic setting. Therefore, I chose to use the standards only when specifically needed (such as when choosing materials, understanding outgassing, or more technical issues), rather than applying the full SE and PM approach recommended. To a certain degree, this approach coincided with the recommendations in the ECSS IOD tailoring, which the team consulted for tailoring the overall development approach. The approach to testing is described differently in the cases, in the HYPSO project, Aalborg, Iris CubeSat and Unnamed project, an approach using early prototypes is adopted. Furthermore, the HYPSO project, Iris CubeSat and Unnamed project all mention using FlatSat testing of software on relevant hardware. In addition, the HYPSO team adopted using MBSE for some of the SE activities, such as requirements management and traceability to design. We also did some of the dependability work using MBSE, to increase visibility of impact of design choices and risks.

Table 4.5: Case study characteristics — I.

Item	Aalborg [28,51,237]	Delfi-C3 [60,237,238]
Country	Denmark	The Netherlands
Development time	Early 2000 - today.	Early 2000 - end of 2008.
Team size	10-30 people.	More than 30.
Team composition	Undergraduate and graduate level students.	Undergraduate and graduate level students.
Project commitment	Students join for a curricular project and/or volunteer. Focused around a project based learning program.	Specific assignments (individual) for thesis or internship work.
Academic accreditation	Yes and volunteer.	Yes.
Funding	External funding and sponsorship from companies. Later satellites partially funded by ESA Fly-Your-Satellite program.	Hosted demonstration technologies from institutions to support cost. Some sponsorship from companies. Supported by national space agency.
Faculty support	Experienced faculty (1-2 dedicated 10-20% plus 5-8 staff members supporting). No PhD students or research assistants. Supervision from a scientific staff member.	Supervisory support for students. Support for technical development.
Experience	More than 3 satellites.	More than 3 satellites.
Workforce flowthrough	Core team that lasts multiple semesters, and changing of members each semester.	Handover between each year (thesis).
In-house facilities	Cleanroom. Workshop. Project rooms. PCB facility. Ground station.	Cleanroom. Workshop. Project rooms. PCB facility. Ground station. Thermal vacuum chamber access. Vibration table access.
Disciplines	Mainly electrical and information technology.	Various engineering disciplines.
COTS/in-house	Mainly in-house development of subsystems.	Mostly COTS approach.
Management	Students are encouraged to do all the leadership and building of the satellite themselves. Faculty is there for support and guidance, not control.	Dedicated project manager. Adopted many space engineering standards for PM.
Communication channels	Not described.	Not described.
Documentation	Not well described, except for design reports at milestones.	Not well described, except for design reports at milestones.
PM approach	Mostly for scheduling. Self-organizing teams.	Structured approach to PM with work packages, phasing, scheduling, documentation management.
SE approach	Requirements, system budget, and interface management.	Adoption of many space engineering standards for SE. Requirements and system budget management.
Testing	Early development of prototypes for testing.	Late delivery of hardware.

Table 4.6: Case study characteristics — II.

Item	Iris CubeSat [78, 239]	Unnamed CubeSat [83, 240]
Country	Canada	Russian Federation
Development window	2018-2022 (present)	ca. 2018-2022
Team size	10-20 people (not clear).	Not described.
Team composition	Not clear, assume undergraduate and graduate students participation.	Payload development at university, supported by enterprises and an institutional partner. Assume undergraduate and graduate level students.
Project commitment	Not described if volunteer or credit-seeking.	Not described if volunteer or credit-seeking.
Academic accreditation	Not described.	Not described.
Funding	Sponsered by the Canadian Space Agency through their CubeSat program.	Not described. Hosted nanosatellite by other parties.
Faculty support	Support from researchers and professional engineers.	Not described in detail. Assume support from researchers and professional engineers in consortium.
Experience	First satellite.	Not described.
Workforce flowthrough	Not described.	Not described.
In-house facilities	Thermal vacuum chamber. Cleanroom. Vibration table.	Not described. Assume support from consortium.
Disciplines	Not given in available documentation. Assume mix of engineering disciplines.	Optics, mechanics, electronics, software.
COTS/in-house	Mix of in-house subsystems and COTS subsystems for satellite bus. Payload developed in-house.	Developed payload for a host nanosatellite mission.
Management	Work lead by PhDs/faculty.	Not described.
Communication channels	Daily "huddles" stand-ups. Formal design reviews, also with institutional partner.	Daily stand-up, sprint review, sprint retrospective. Formal design reviews.
Documentation	GitHub for software version control. All documentation available on team google site (similar to wiki).	Atlassian Jira to keep track of issues and tasks for kanban. Assume formal documentation for collaboration with partners.
PM approach	Self-organizing team. Agile with kanban board and gantt. Monitor progress according to verification activities.	Agile hardware development.
SE approach	Requirements management with cloud tool. Progress followed with a verification burn chart. Integrated to overall PM approach in agile.	Formalized requirements and ICD process with partners. Iterative deliveries for hardware.
Testing	Unit and FlatSat testing of software.	FlatSat testing. Multiple rapid prototypes and assembling for hardware testing.

Table 4.7: Case study characteristics — III.

Item	**HYPSO CubeSat**
Country	Norway
Development time	2017 - 2021 (present)
Team size	20–30 people.
Team composition	Undergraduate and graduate students. 6-8 PhD/Post doc.
Project commitment	Thesis work (bachelor/master/PhD). Research projects (Post doc.). Bachelor: group assignment. Master: individual assignments.
Academic accreditation	Yes.
Funding	Mainly funded by the Norwegian Research Council and NTNU, support from various companies, and some development funding from the Norwegian Space Agency.
Faculty support	Supervisors for thesis work. One electrical engineer (part-time). One purchaser (part-time).
Experience	First satellite.
Workforce flowthrough	Students join for 1–2 semesters. PhDs join as a part of their research.
Disciplines	Mainly electronics, cybernetics. Some mechanics and optics.
COTS/in-house	Satellite bus COTS. Payload developed in-house with COTS components.
In-house facilities	Thermal chamber (no vacuum). Access to cleanroom. Electronics and mechanics workshop. 3D-printer. Project area. Ground station
Management	PhDs as project manager and subsystems leaders.
Communication channels	Slack for informal day-to-day discussions. Shared office space. Daily standups. Weekly management meetings. Semesterly design reviews.
Documentation	GitHub for software and electronics version control (connected to kanban issues). Design review formal documentation that can be included in project reports/thesis written in cloud so many people can work on together. Automatic doxygen from Jenkins documentation when software update. Digital platform for design review. MBSE for requirements traceability to components, verification, dependability analysis.
PM approach	Agile on software, hardware and AIT. Integrated with Git kanban boards. Thesis tasks and project tasks in kanban together.
SE approach	MBSE for requirements traceability to components, verification, dependability analysis.
Testing	Unit and FlatSat verification of software. Multiple development models to verify and validate functionality quickly.

4.6.1 **Case Study Description**

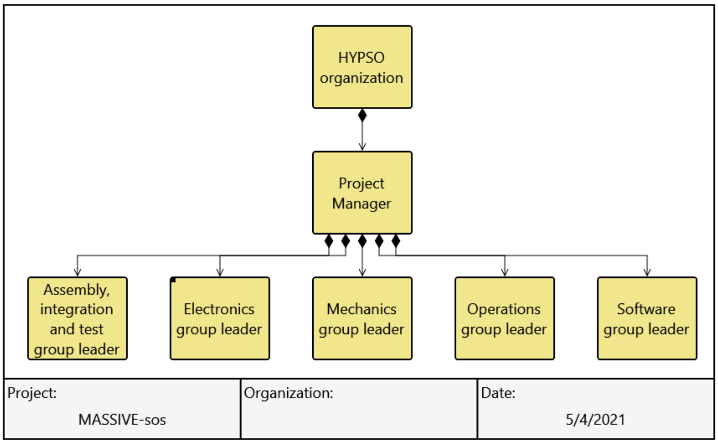


Figure 4.7: HYPSO-1 project organization with group leaders. The mechanics group includes both optical and other mechanical hardware. Note that no systems engineer is identified. This role was not fully staffed, but the project manager took on some of the tasks and tried to follow up as much as possible. Diagram made using GENESYS from Vitech corp.

The primary case study is the HYPSO project, a satellite mission funded by the Research Council of Norway (RCN), the Centre for Autonomous Marine Operations and Systems (AMOS), and supported by the Departments of Electronic Systems and Engineering Cybernetics at NTNU. The project description is given in Section 1.1.1, and Figure 4.8 provides a breakdown of the larger system.

Table 4.8: HYPSO team composition per year. PD refers to Post Doc. researchers. Mech. refers to studies associated with mechanical engineering. Elec. refers to electronic engineering. Cyb. refers to cybernetics.

Year	Mech.		Elec. Systems		Eng. Cyb.		Others	
	BSc/MSc	Phd/PD	BSc/MSc	Phd/PD	BSc/MSc	Phd/PD	BSc/MSc	Phd/PD
'17/'18			0/6	3/1		2		
'18/'19	6/3		0/6	3/1	4	4		1
'19/'20	0/1	1	5/7	2/2	3	6	0/4	1/1
'20/'21	3/3		0/6	3/2	4	6		1/1

The HYPSO project team consists of approximately 20 BSc and MSc students and 6-8 PhD or higher-level employees, roughly distributed in different disciplines. The group composition is given in Table 4.8. Most of the students are Norwegian, and few have previously participated in multidisciplinary projects. The group leaders are all PhD candidates and "report" to the project manager, also a PhD candidate, shown in Figure 4.7. A use-case diagram of the project is shown

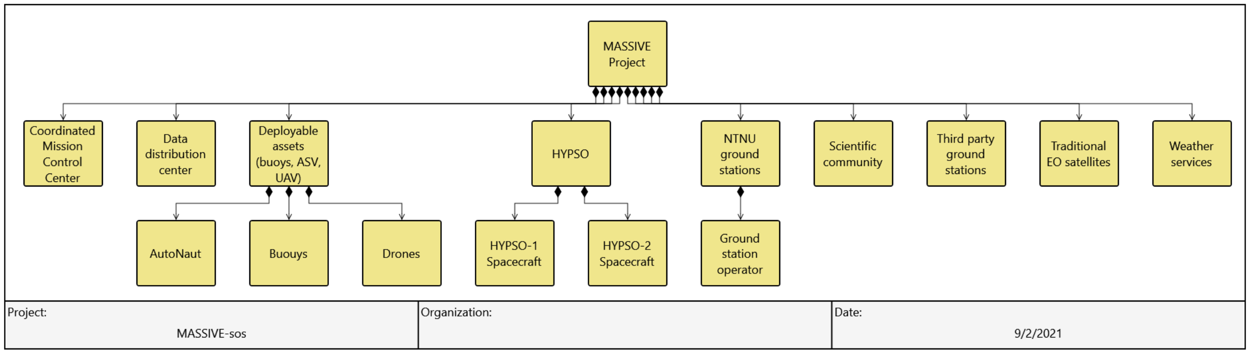


Figure 4.8: MASSIVE project organization with its constituent systems. Diagram made using GENESYS from Vitech Corp.

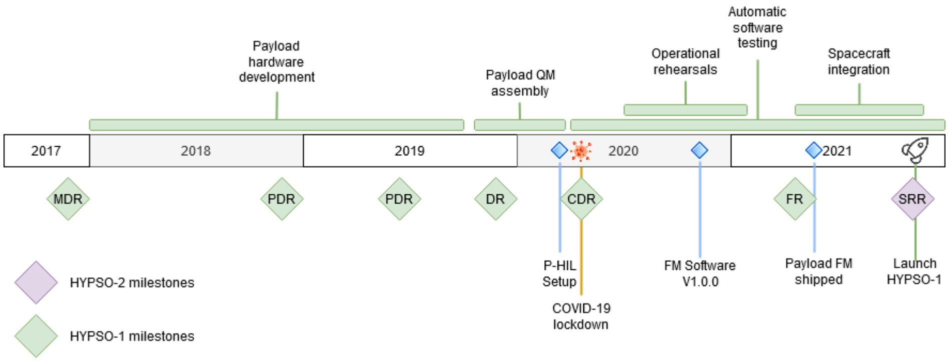


Figure 4.9: Overview of HYPSO-1 development and milestones.

in Figure 4.10. Furthermore, there are about 10 professors closely or loosely associated with the project who have different needs and expectations for the project and its execution. One of the challenges with the HYPSO-1 project team was that few or none of the members had prior experience in designing and building CubeSats, neither the professors nor the students. A timeline of the HYPSO development is given in Figure 4.9. Every summer, the team of students change.

System Boundaries Identifying the system boundaries limited the scope such that the problem would be solved within the time allocated. The context diagram is shown in Figure 4.11. The process shown in Figure 2.2 limited the scope of this research to looking at methodologies to support the satellite development phase within an academic environment. However, the results and experiences gained are relevant in a broader context.

Stakeholder Analysis To determine the appropriate methodology and processes used in small satellite development, the relevant stakeholders and their needs were identified. The stakeholders were determined by the context of the system-of-interest shown in Figure 4.11 and were classified as primary or secondary based on their level of involvement in the system. The stakeholders and needs are given in Table 4.9.

The “NTNU: MASSIVE project” represents the project owners who were the source of funding for the satellites. The “NTNU: PhD supervisory committee” are the supervisors of the PhD candidates working in the research group and the committee at the departments concerned with the quality of PhD research and the scientific results. The “Satellite system designers (internal)” refers to the group of PhD candidates and master students working on the design and development of the payload, including the integration to the spacecraft. Following this, the “Satellite system designers (external)” refers to anyone external to the case studies concerned with this research. The “Suppliers” refer to suppliers that provide the spacecraft itself, ground support equipment, and other services related to the spacecraft operations. The “Research Council of Norway” is the funding agency of the MASSIVE project and is influenced by national political strategies. The “Launch providers” are identified as a tertiary stakeholder because the regulations regarding launch safety strongly influence the documentation requirements for the payload.

For the whole satellite development lifecycle, the stakeholder picture was more extensive, shown in Figure 4.12. Many actors influence requirements, decision-making, and design. The problems occur when current design information is not communicated to all parties with objections or supplemental knowledge to improve the design. For instance, the "HYPSO-1 project organization" cannot decide on a communications band without discussing it with "International Telecommunication Union (ITU)" first, and they must ensure that there is space for the radio and antenna on the satellite as well as available equipment (from "suppliers").

The "Satellite system designers (internal)" were identified as primary stakeholders because they will ultimately be the ones who use the methodology. They stated that usability was one of the most critical requirements because the participants are already pressed for time and do not want to spend too many resources learning new methods or tools. Furthermore, spending time documenting the design and ensuring traceability was not their primary focus. The designers wanted to spend their time building the satellite.

The other primary stakeholders were related to NTNU management: the MASSIVE project owners and the PhD supervisory committee. While space-related activities at NTNU are done at both master and PhD levels, the funding came from the MASSIVE project and the PhD candidates were the ones carrying out the research activities. On the one hand, NTNU wanted all PhD candidates to finish on time and provide high-quality research. On the other hand, the MASSIVE project wanted to develop and launch two satellites within the time frame given by the RCN to ensure their standing with the funding body. The project owners also had requirements for enabling transparent communication of project status and technology development reports.

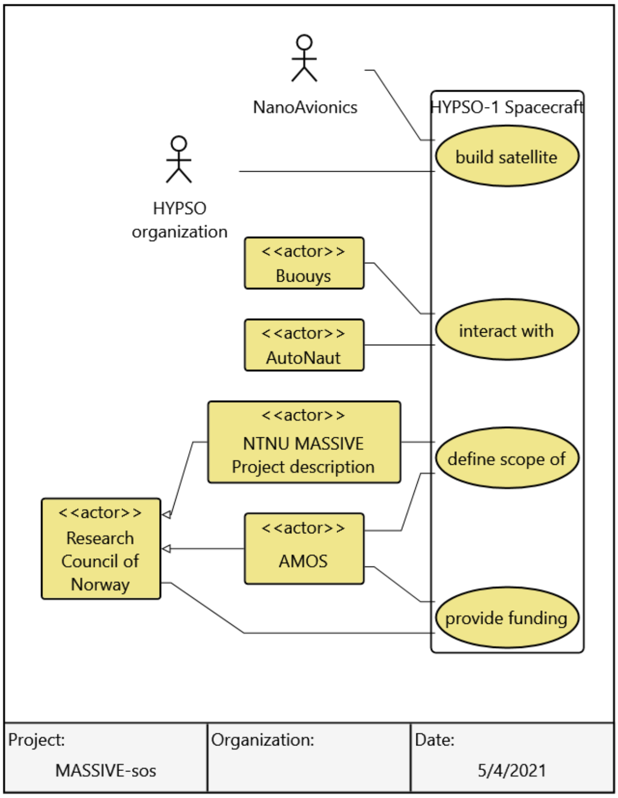


Figure 4.10: Building the HYPSO CubeSats as a use-case diagram. Diagram made using GENESYS from Vitech corp.

Table 4.9: Satellite system stakeholders.

Stakeholders	Involvement	Needs
NTNU: MASSIVE project	Primary	Ensure rapid and reliable development of small satellite systems. High efficiency in documentation to ensure transfer of knowledge with minimal effort. Interested in frameworks that can support this, as well as low-cost methods. Shorter development cycles and iterations.
NTNU: PhD supervisory committee	Primary	Ensure that all PhD candidates graduate with high-quality research.
Satellite system designers (internal)	Primary	Usability of software tools and methodologies, simple to understand, multi-platform. Exchange of information across subsystems and traceability between design documentation, requirements, models, and test results.
Satellite system designers (external)	Secondary	Same as above, as well as open source.
Suppliers	Secondary	Exchange of models and requirements in a simple format. Usually a known, standardized format.
Research Council of Norway	Secondary	Fulfill the goals of the research project MASSIVE with quality results in autonomous marine operations.
Launch providers	Tertiary	Exchange of models and requirements in a simple format. Usually a known, standardized format. Clear compliance to launch regulations, especially safety regulations.

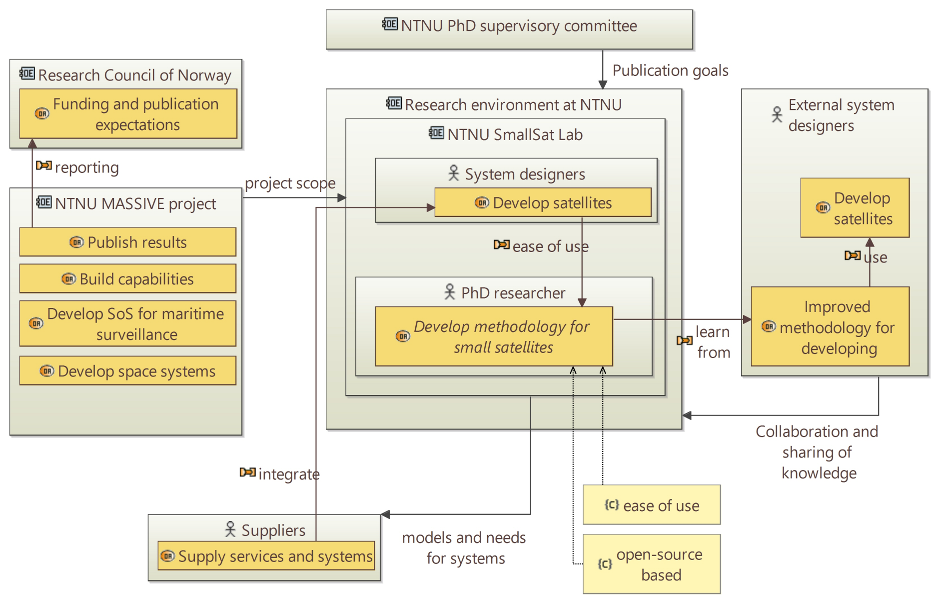


Figure 4.11: Context diagram showing stakeholders and the system of interest in the larger system for this PhD research. Diagram made using Capella from the Eclipse Foundation/Thales group.

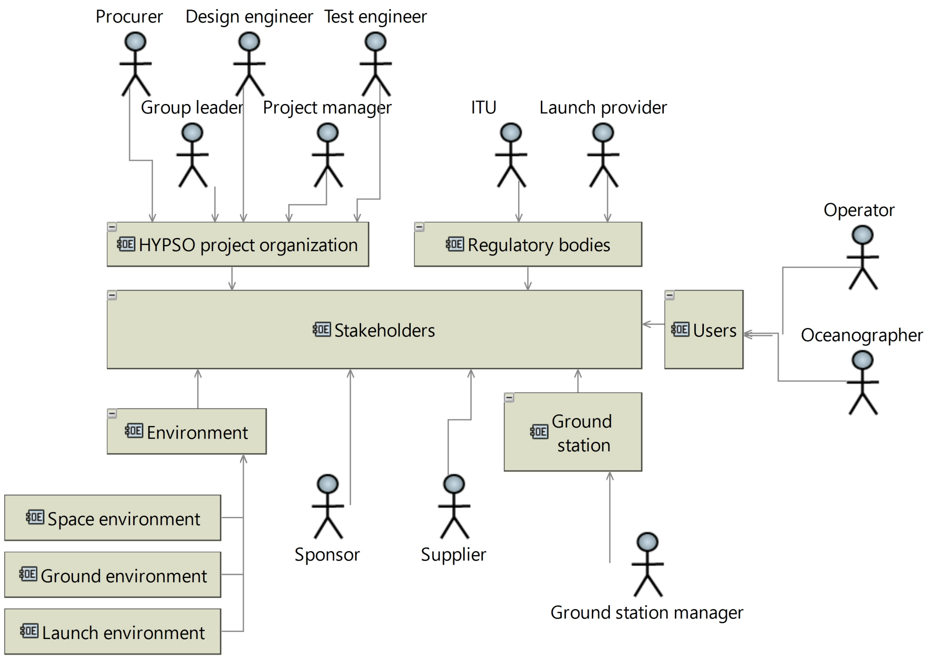


Figure 4.12: Stakeholder overview for the spacecraft. Diagam made using Capella from the Eclipse Foundation/Thales group.

4.7 Tools used for Analyzing the HYPSO Case Study

In this section, the methods that are used in the HYPSO project case study are presented.

4.7.1 Agile Decision-Guidance Method

The Agile Decision-Guidance Method (ADGM) is based on the Observe-Orient-Decide-Act (OODA) loop developed by John Boyd [241]. The method came out of the INCOSE Agile Systems and Systems Engineering Working Group to respond to the needs of several stakeholders and brainstorming sessions. It has three observation spaces: *the customer problem space*, *the solution space*, and *the product development space*, as well as a *response capability assessment*, which are evaluated through thirty-six factors and over 100 questions. The results are given in a spreadsheet, with graphs showing the stability, variety, observability, and predictability of a factor. The charts are used to help management teams decide where to focus efforts for increasing agility. An in-depth description of the method can be found in Lyells et al. [241].

Application of ADGM ADGM was used as a discovery method for the Action Research, and was applied in 2020 and 2021 through online meetings facilitated by (i) "Jane's story" (a method from ADGM, see Appendix 8) and (ii) a questionnaire developed by Lyells et al. [241]. I wrote the initial draft of "Jane's story," which was subsequently reviewed by the HYPSO team leaders and can be found in Paper F [17]. The HYPSO project was evaluated by the questionnaire from April 2020 – June 2020, with 2–4 members of the HYPSO group leaders. In the fall of 2020, the HYPSO group leaders discussed potential outcomes and ways forward with the authors of ADGM [241] by analyzing the outcomes of the questionnaire together with the notes and "Jane's story."

4.7.2 Systemigram

A brief description of systemigrams and how to use them is included since it was one of the main techniques used for modeling the sociotechnical systems in this research. An example is shown in Figure 4.13. The systemigram was introduced by Boardman [242], a "SYSTEMs dIaGRAM", first used to support systems engineers and project managers understand a system in its context. It is a network of nodes and links, with a specific flow from beginning to end. Sauser et al. [242] list several papers where systemigrams have been demonstrated. Examples of applications where systemigrams have been applied include disaster relief management, organizational learning platforms, communication bottlenecks, and cybersecurity policies. The "rules" for creating a systemigram are as follows [242]:

1. The systemigram should be read from the top left corner to the bottom right, which should be the final goal or output of the system.
2. There should be 15-25 nodes.

3. It includes nouns representing people, organizations, systems, or other artifacts. These relate to verb phrases that should “indicate transformation, belonging, and being [242, p. 281].”
4. No repeating nodes.
5. Link crossover should be avoided.
6. Coloring and “beautification” of nodes and links can increase the clarity.
7. The topology of the diagram should enhance the expression of *why, how, and what*.

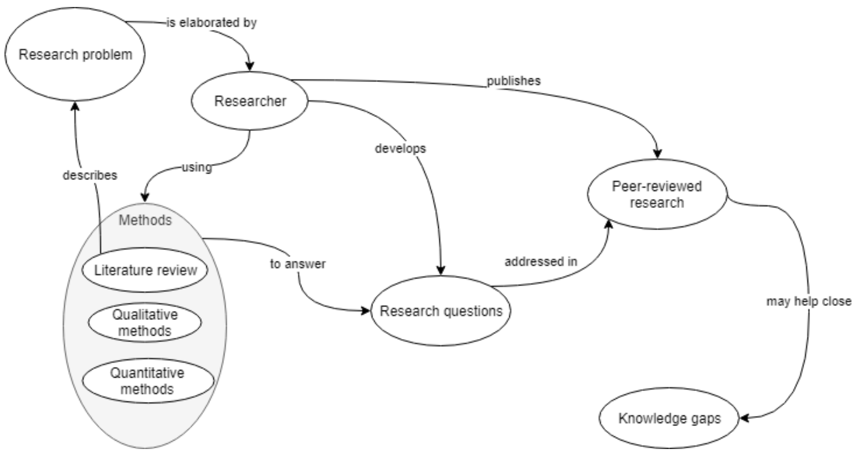


Figure 4.13: Example of a systemigram.

Application of Systemigram The systemigrams were used to model different aspects of the sociotechnical system of the HYPSO project team. These were used in team meetings and discussions to understand the dependencies and mechanisms between the elements in the system. The systemigrams went through several iterations and revisions with team members until there was an agreed-upon understanding that it represented how people saw the sociotechnical system. They were also used to inform stakeholders about the complexity of the organization.

4.7.3 **N2-diagram**

The N-Squared Chart, N2-matrix, or N2-diagram is “in the shape of a matrix, representing functional or physical interfaces between Systems or System Elements. It is used to tabulate and analyze the interfaces [243, p. 31].” The N2-diagram can be used for both functional and physical interfaces. The system components or functions are mapped diagonally in a matrix, and the remaining cells represent the interfaces.

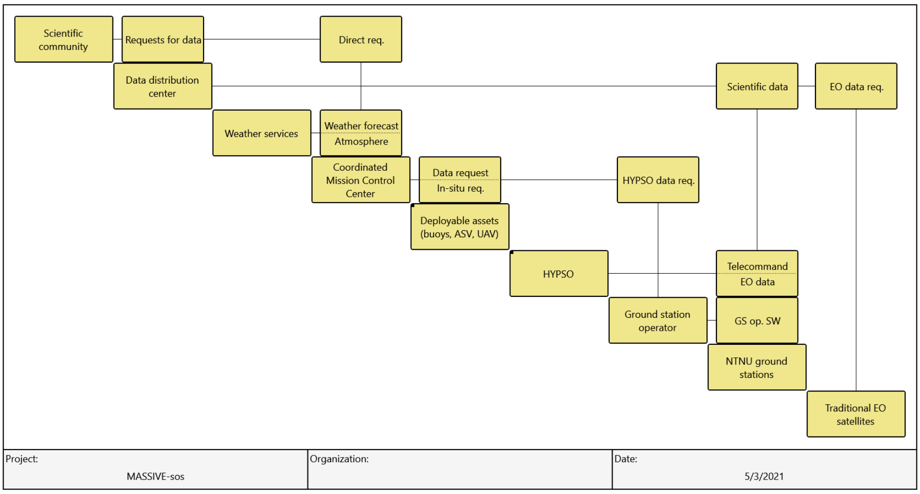


Figure 4.14: The N2-diagram from the MASSIVE Project.

Application of N-Squared Chart

The N2-diagram has been used for supporting the interface analysis for estimating the System Readiness Level (SRL), where the subsystems were sorted in “internal subsystems”, “environments”, and “external subsystems” [14]. The N2-diagram has been used for helping team members understand the interdependencies of their assignments when developing the HYPSO CubeSat [12] see Figure C.1. It has also been used for understanding the interfaces in MASSIVE from an SoS perspective to develop operational scenarios and interface descriptions, see Figure 4.14.

4.7.4 Technology Readiness Levels

While the availability of COTS components has contributed to the popularity of small satellites, there are challenges with interoperability between components, obsolescence, and verification and validation of integrated systems [244]. Many CubeSat projects combine COTS subsystems with in-house developed components. Furthermore, many COTS component manufacturers may not provide the complete datasheets describing the behavior of materials or processes used to manufacture the components. This lack of information could lead to unexpected challenges when integrating elements from different vendors into a spacecraft. ESA and NASA have standards for qualifying COTS components for their space missions. These are often too costly for academic institutions or small companies, requiring extensive testing, documentation, and production of a batch of components to qualify the single flight component [245]. Small satellite developers must use a different approach to verify that their choice of COTS components is suitable for their space mission.

The ECSS defines "readiness levels" to measure the maturity of systems, which can be used to make informed estimations of schedule and work needed to reach completion [246]. The "original" readiness level is called the Technology Readiness Level (TRL), which was introduced in the 1980s [247, 248]. CubeSat subsystem vendors often use TRL in their product data sheets to inform potential customers of the qualification status of their subsystems, and providing a maturity level assessment may be required by funding bodies or when applying for projects [246, 247]. In Bakke and Haskins [249] the authors provide a thorough review of the TRL and its application the past decades, as well as the limitations of using the TRL, such as *Integration and system views* and *Capability ("the system's ability to produce an operational outcome [249, p. 6],"* which are not included in the original TRL. The use of the component in its intended context and system affects how ready the component is [247]. This led to the introductions of Integration Readiness Level (IRL) [250–252] and System Readiness Level (SRL) [247]. For the CubeSat projects, when spacecraft developers use many subsystems from different vendors when designing their system, the application of Integration Readiness Level (IRL) is critical. According to the literature, many of the issues university CubeSat developers encounter are related to integration, especially at a system level [24, 51]. As such, CubeSat developers should not only rely on TRL when developing complicated systems such as spacecraft because the subsystems need to function together, they should also recognize the importance of IRL and resulting SRL [247, 253].

According to [246], there are still limitations in applying the SRL to research and development projects or when combining multiple systems. Suggested improvement include using a nested model as described in [254], with the inherent challenge of deciding on system boundaries when making component SRLs to use in the nested SRL. In Olechowski et al. [248], the limitations of using SRL stress that the IRL and TRL appear to be independent while in fact they are dependent. Furthermore, in Kujawski [255] it is outlined how IRL and TRL are ordinal numbers, and that the matrix multiplication used for the Sauser SRL is not valid, which results in a distortion of the system readiness and can lead to wrong interpretations. Another limitation is how IRL and TRL are assessed by *subject matter experts* supported by comparison to previous work, established standards, and *gut feelings* [256]. This is also discussed in [249], highlighting that a person's background and organizational culture will influence the assessment. Miller et al. [257] also introduced a framework for accounting for the human element in a system through the "Human Capability Level" and "Human Integration Readiness Level".

Application of the Readiness Levels The readiness levels were used in [14] to investigate if they can be a useful tool for managing and prioritizing tasks in university CubeSat projects. The efforts of estimating the readiness levels, although based on the subjective evaluations of team members, can be useful since the same people evaluate all the subsystems and can rank them relative to each other. However, when team members change, we should be careful comparing evaluations at different times to make decisions.

4.8 Software Tools Used

The tools described in this section were used throughout to support the activities related to the Action Research.

4.8.1 Model-Based Systems Engineering Software

I used two software programs for MBSE purposes. The first one is CORE (and the newer version GENESYS) from Vitech Corporation[2], which were provided through a research license. The second one is the open-source Capella tool with Arcadia method[3]. The two tools use different approaches and have different relational metamodels underlying their MBSE. GENESYS/CORE was mainly used for requirements engineering and system modeling of the HYPSO CubeSat because of its metamodel and support for traceability to verification requirements and events. Capella was primarily used for MASSIVE systems modeling and operational scenario modeling because it is open source and more developers could use it simultaneously.

4.8.2 Systemigram Software

The systemigrams were developed using SystemiTool[4]. The tool offers visualization enhancements to engage stakeholders by highlighting small areas of interest in the larger systemigram. For example, in this thesis work, the systemigram was used when discussing how a student project works, how the system information was structured and used, how student responsibilities differed but were co-dependent and how the stakeholders affect the work and the organizations. The systemigram discussions were facilitated either by having online virtual meetings with a shared view of the whole systemigram or going through it from left to right using the SystemiTool.

4.8.3 ECLIPSE Configuration Management Tool

The software-as-a-service application ECSS Compliant Toolset for Information and Projects Support of Enterprises in Space (ECLIPSE)[5] was used for managing tasks as reviews, document configuration, action item control and risk management. The software was provided through an educational license, and the Sapienza team held a training session for the HYPSO project team. ECLIPSE is a common tool for companies in Europe participating in ESA projects and follows the ECSS configuration management and risk management standards. The HYPSO team members were given some in-house training and an external workshop on how to use ECLIPSE, but had no prior experience using the system before joining the project. I had used it in a previous workplace in an older version.

[2] Web page: `https://www.vitechcorp.com/`.
[3] Web page: `https://www.eclipse.org/capella/`
[4] Web page: `https://sercuarc.org/serc-tools/`.
[5] Web page: `https://www.eclipsesuite.com/`

4.8.4 Tools used in the HYPSO Project

The software tools used by all student members, PhD and Post-Doc. researchers in the HYPSO project that are mentioned in this thesis are given below. Most HYPSO team members had prior experience using Google Drive, but not GitHub/Slack/Zoom.

- Slack[6] used for direct messaging and project group channels.
- GitHub[7] used for agile project management, code repository, and version control with educational license.
- Google Drive[8] used as shared document drive for working documentation.
- Zoom[9] used as video conference tool.

[6] Web page: `https://slack.com`
[7] Web page: `https://github.com/`
[8] Web page: `https://www.google.com/drive/`
[9] Web page: `https://zoom.us`

Chapter 5

Summary of Appended Papers

The more clearly we can focus our attention on the wonders and realities of the universe about us, the less taste we shall have for destruction.

Rachel Carson

The three research questions look at different aspects of improving how we work to deliver systems for space and integrating systems to provide mission capabilities. This is a sociotechnical challenge, where the people developing the systems are critical to ensure project success. It is challenging to develop and integrate systems that combine in-house developed and COTS elements because documentation is not always fully transparent or fails to give the necessary information.

This chapter provides a summary of the appended papers in this thesis. They cover different aspects contributing to the understanding of applied research projects in academia, specifically space projects. An executive summary of each is provided here for reference. The findings are discussed in light of the research questions in the next chapter.

Paper A describes the findings from the first year of the HYPSO case study related to team management, design reviews, and design process [12].

Paper B demonstrates the application of reliability methods for implementing dependability analysis in MBSE and shows how this can benefit CubeSat teams struggling with limited personnel resources and low experience with space systems [13].

Paper C introduces factors that influence the development time for university CubeSats, and provides an analysis of the impact of the factors with respect to which lifecycle phase of the system is in [14].

Paper D describes how the HYPSO project fared during the COVID-19 lockdown, and how the introduction digital collaboration tools require new considerations to systems engineering and project management for the university CubeSat project [15].

Paper E builds on Papers A, B, C, D, F, and G, and sets out the Digital Engineering approach applied by the HYPSO team, and presents how the systems engineering and project management activities are integrated into the workflow and digital engineering practices [16].

Paper F presents the analysis of the HYPSO organization using the Agile Decision-Guidance Method (ADGM), how effective it was for academic CubeSat organizations, and the identification of which areas of the organization could benefit from an agile approach [17].

Paper G narrates the challenges of sustainable management of arctic coastal regions, and provides an analysis of how the MASSIVE SoS can contribute to addressing identified stakeholder concerns and needs [18].

Paper H uses the knowledge from Paper G and a MBSE approach to elicit architectural concepts and technical considerations for a SoS with a small satellite and autonomous surface vessel for persistent coastal monitoring [19].

Paper I presents the results of 18 semi-structured interviews looking at academics' perception of systems engineering and applied research projects, to contribute to the understanding and management applied research projects in academia [20].

Paper J builds on the same interviews as Paper I, as well as a literature study of complexity, to investigate how complexity language, with roots in both project management and system engineering disciplines are reflected when informants talk about the role of project management and systems engineering in two academic organizations. The paper continues to contemplate how these findings inform efforts towards a unified project complexity language for the two disciplines [21].

5.1 Paper A: CubeSats in University: Using Systems Engineering Tools to Improve Reviews and Knowledge Management

The first paper published as a part of this PhD reported on the findings from 1.5 years of the case study. It describes the HYPSO project and milestones, outlining key challenges related to team management, design reviews, and the design process. The HYPSO project team experiences a break in continuity at the end of each school-year when master and bachelor students graduate. The team members joining in the fall may not have experience with space systems or with multi-disciplinary team work and what it means to develop a spacecraft with so many dependencies. The systemigram shown in Figure 5.1 was developed to enable better communication within the team about how the project works, and with stakeholders so they can understand how the team experience the project. Until the fall of 2018, the HYPSO team was distributed between different working offices and only met once during the week at the weekly meeting. Learning from other CubeSat teams, it was decided to adopt some of the Concurrent Engineering practices by moving the team into a common working area, introducing a shared cloud storage area and concurrent working times, and increasing informal communication meeting points. The team now has more frequent information exchanges, more open body language, and overall better information sharing. Furthermore, an N2-diagram was used to show how team members' tasks are related to each other helped people understand how their work could influence others and vice versa. This is shown in Figure C.1.

Design reviews function as both a stage-gate for moving between development phases, but also as a point-in-time where the whole team is aligned around a common design and mental model of the system. Additionally, design reviews are when developers make the extra effort to make the documentation available for external reviewers, and thus, contributing to the information system of the project. The evolution of HYPSO design review from Mission Design Review (MDR) to Preliminary Design Review (PDR) is described. The design review process applied in HYPSO project is shown in Figure 5.2. Participants at MDR included university faculty members (in radio communication, autonomous systems, ADCS and control, oceanography) and 4 PhD students. Feedback from MDR was collected orally during the meeting, in addition to a lessons learned session later from which a minutes-of-meeting was written. The results and inferences presented in the paper from the MDR feedback were reviewed by participants in the MDR. Participants at PDR included the HYPSO team at the time (students and PhDs), faculty members, representatives from the Norwegian Space Agency with experience in CubeSats and satellite programs in general, representative from external partner in on-board processing, representatives from NTNU student CubeSat organization, and representative from NanoAvionics (the satellite bus provider). The participants were invited to provide feedback continuously during the PDR meeting, and these were partially collected in minutes of meeting. Then, the feedback was summarized in the PDR review report which the HYPSO group leaders contributed to. Feedback from PDR includes: (1) the PDR form helped in providing feedback; (2) too little time to review; (3) clearer description

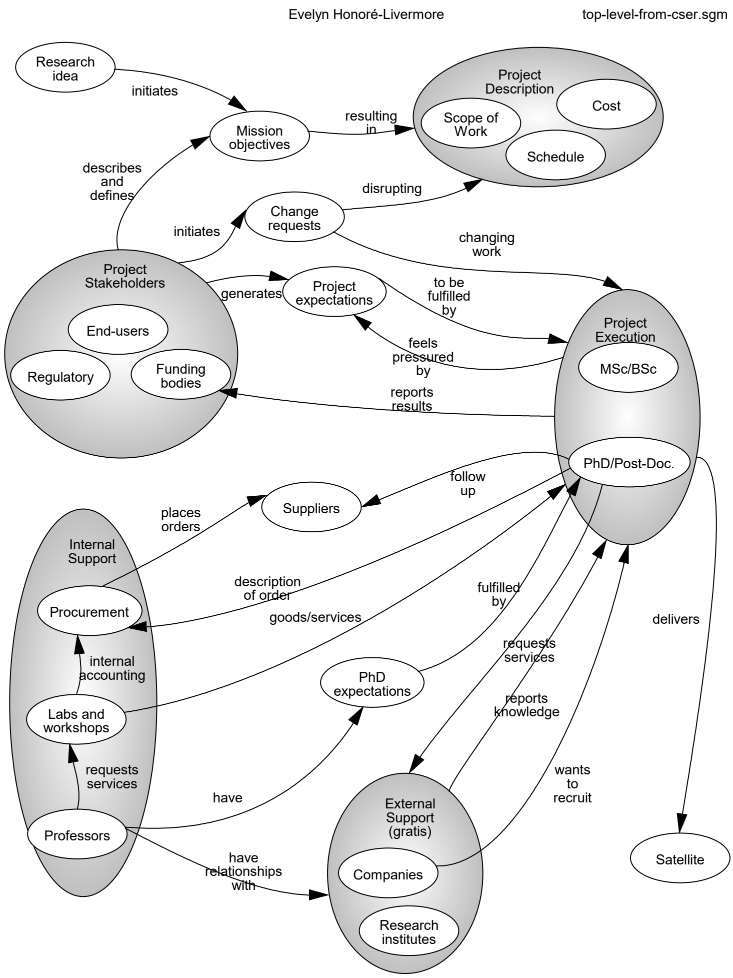


Figure 5.1: The first systemigram (adapted from the publication to follow the same layout as with subsequent diagrams using the SystemiTool). The Project stakeholders are looking for higher performance and faster product delivery, the Internal support are interested in seeing the Project execution team succeed with both the theses and the project goals (at the same time), and the funding bodies RCN, Norwegian Space Agency (NSC), and NTNU want everything to be completed and published on time and on cost. The Project execution wants both to fulfil the expectations for their theses to finish their formal tasks on time to finish their degrees, and to build a satellite that satisfies the Project stakeholders.

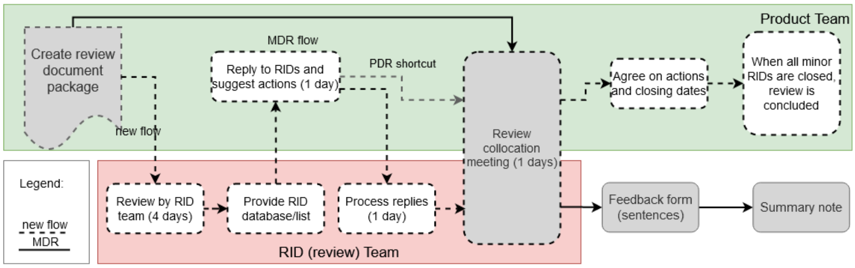


Figure 5.2: The design review process based on ECSS standard. The two boxes "Feedback form (sentences)" and "Summary note" are not a part of the standard, but were used for the Mission Design Review (MDR). Until Critical Design Review (CDR) of March 2020, all collocation meetings were in-person. Since then, they have been virtual.

of mission/be invited to a mission review; (4) poor traceability of spacecraft requirements; (5) better structuring and overview of documentation would have helped when reviewing; (6) two day collocation (one day to go through Review Item Discrepancy (RID) and one day to work on technical discussions/workshops. Based on this feedback, the review process was iterated on, and subsequent reviews now include a presentation to address points 3 and 5; split in two half-days to address 6; clearer inclusion of requirements and traceability with an Assembly, Integration, and Test (AIT) plan and verification plan to address 4. Getting feedback from the review time in time is challenging, because people are doing it on a volunteer basis and often do not have enough time amongst other tasks (ref. point 2). Spending time during the review to discuss technical issues generates more RIDs, which is helpful for the design moving forward.

5.2 Paper B: Model-Based Systems Engineering for CubeSat FMECA

A part of having a successful mission is ensuring a dependable system, which can be challenging when using COTS components and the knowledge management associated with high turnover. Too much rework also influences the development time, and if the efforts to ensure dependability is time-consuming, or not performed as an integrated part of the design lifecycle, it could affect the schedule negatively. This is addressed in the framework of using Model-Based Systems Engineering (MBSE) incorporating Reliability, Availability, Maintainability (RAM) models to improve the design concept. The background for studying this is how RAM activities are often done separate from the design concept development, which the project team want to circumvent by actively introducing RAM activities early in the design. The project team implemented the RAM results in the MBSE system model, and integrated the RAM engineers into the design team so that the other designers would consider the RAM results in their work. A qualitative Failure Mode, Effects, and Criticality Analysis (FMECA) is presented, and these results were included in

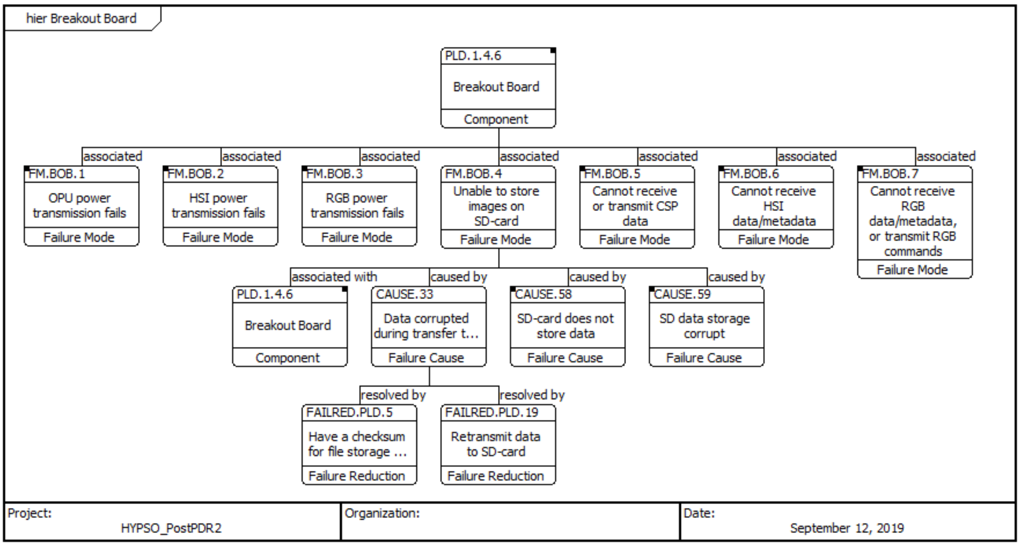


Figure 5.3: A subsystem (BOB = BreakOut Board) shown with its failure modes displayed in a hierarchy diagram. One of the failure modes has been expanded to show the failure cases and reduction methods. Model using CORE from Vitech corp.

an extended MBSE system model with new classes introduced into CORE (Vitech Corp.) based on the [197] presentation. An example of this is shown in Figure 5.3, where suggested failure reduction measures are given. More of the model can be found in Appendix 8. The failure reduction measures could then be associated with actions and assigned to developers. The project team found that having the FMECA as a part of the MBSE system model increased the communication on RAM considerations for design, and resulted in among other things, functional and physical design changes. Furthermore, the work associated with this effort is not too large, and much can be reused for future satellites with similar components and architecture.

5.3 Paper C: Factors Influencing the Development Time from TRL4 to TRL8 for CubeSat Subsystems at a University

The findings in this paper are based on a mix of short interviews from different members in the team, and evaluations from the group leaders. The group leaders are four PhD fellows and one engineer, and have a mix of background from electronics engineering, cybernetics, aerospace engineering. Two of the group leaders (electronics engineering) have more than 10 years experience in the field, while the others have less than 3 years. The students interviewed were a new software developer (new to the HYPSO team) and a graduated software developer, both working on embedded systems. As the HYPSO project is focused on developing the payload subsystem, we look at both the TRL of the payload, and the SRLs of software and hardware modules in this paper.

The factors affecting development time are: (1) **lab facilities**, (2) **university facilities**, (3) **team knowledge**, (4) **parts supply chain**, (5) **clear objectives**, (6) **internal communication**, and (7) **external facilities**. We find that team knowledge is important throughout the lifecycle, while e.g., university facilities (such as the machining workshop or access to measurement instrumentation), are more important when going from TRL5 to TRL6, because a lot of that effort is associated with qualifying the integrated system in a relevant environment. Furthermore, clear objectives are identified as a source of slower onboarding of new team members. Some of the interviewees thought that it would have been easier to start their projects if they had a clearer scope, objectives, architecture, and requirements defined.

5.4 Paper D: Managing Product Development and Integration of a University CubeSat in a Locked down World

The COVID-19 outbreak affected the whole world during the spring of 2020. Governments reacted by locking down countries and telling people to stay at home, and the university asked the students to work from home. Little previous research was found on managing virtual student teams, especially when not as initially planned. In this situation, many management tasks proved to require more effort than usual, such as managing team members, helping maintain work/home-balance for team members with families or focusing when working from home, and ensuring motivation and on-time project deliveries. The paper presents different aspects of the locked down situation, including technical considerations and managerial considerations. It also discusses the introduction of a HIL setup coupled with a GitHub digital workflow greatly lowered the impact of working-from-home. This is further elaborated in Paper E.

The review process described in Paper A received new attention in 2020, when COVID-19 required a lock-down of the university and the planned CDR was changed to a digital design review. The method of collecting feedback from the review team went from being comments in a meeting and notes on a feedback form, to a spreadsheet based RID form, to a digital cloud-based solution RID database, as shown in Figure 5.4.

The COVID-19 pandemic lockdown provided some unique opportunities to study what modes of interaction worked well and what managers need to consider in such situations. The team went from a situation where most people saw each other every day in the lab to a lockdown in which all interactions took place with digital tools. The results show that having good digital infrastructure is essential, both for system development and team cohesiveness. For the team cohesiveness, frequent Scrum stand-up meetings with video, structured informal conversations, and the Slack channel improved the situation. The importance of having a team culture and work processes is highlighted. The team culture had been instilled in the years prior to the lockdown and was then transmitted from the legacy persons to new team members. Also, the project manager's responsibilities of managing diverse work-from-home situations and communicating deadlines are important.

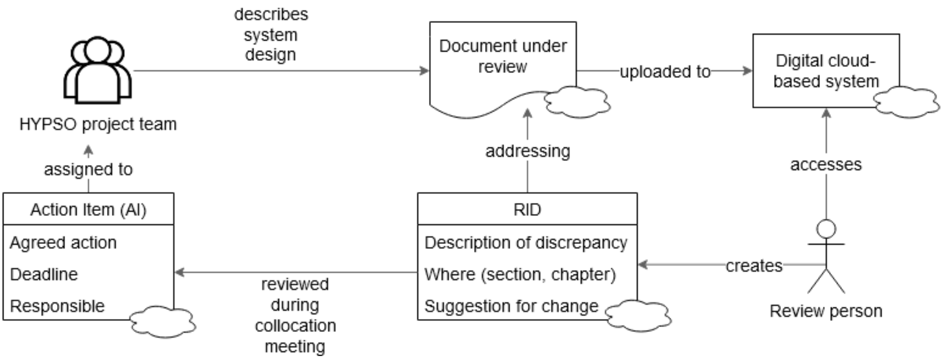


Figure 5.4: Flow of information during design review. The documents are written in a cloud-based software so that many people can work on them simultaneously, and then uploaded to the cloud-based review system.

5.5 Paper E: Digital Engineering Management in an Academic CubeSat Project

Here we look at the project as a whole with a digital engineering lens, and what it means for the future of SE for space development projects. We describe how hands-on learning through an academic CubeSat project could be beneficial for students to gain non-technical professional skills. Additionally, the tailored agile approach is especially suited to projects in academia, where there needs to be a balance between engineering and research tasks, and where we need to have approaches that allow for change depending on the needs of the team. We also include thesis writing tasks and issues related to research questions in the Scrum issues, which could help motivate team members to work concurrently with engineering and research tasks. There are various DE measures incorporated in the HYPSO project lifecycle, and some are shown in Figure 6.4.

We describe the project timeline from inception to today, the software system architecture, and how we tailor the agile methodology to fit the academic situation of the HYPSO team. Furthermore, we describe the HYPSO project approach to verification and validation using a Git workflow and HIL setups. In the paper, we highlight how having an integrated and tailored approach to applying DE methods and tools, coupled with an agile practice to development, can be useful in an academic team. We also show the agile performance over time, as shown in Figures 5.5 and 5.6. The software sprint statistics show how the team used to include many points in a sprint, but learnt over time that it was not possible to attempt that many issues at once. The hardware Scrum approach started a bit later than the software Scrum, so they used that knowledge when choosing how many points to attempt in each sprint, and there is a smaller difference between points attempted and points done. Neither team has reached a level where the number of points attempted equal the number of points done.

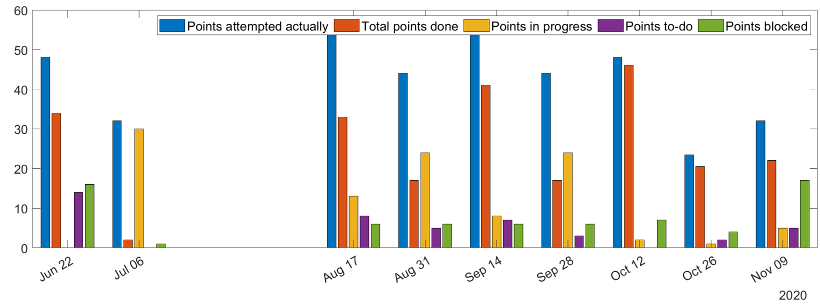


Figure 5.5: Hardware sprint barplot. There was a break of sprints during the summer holiday.

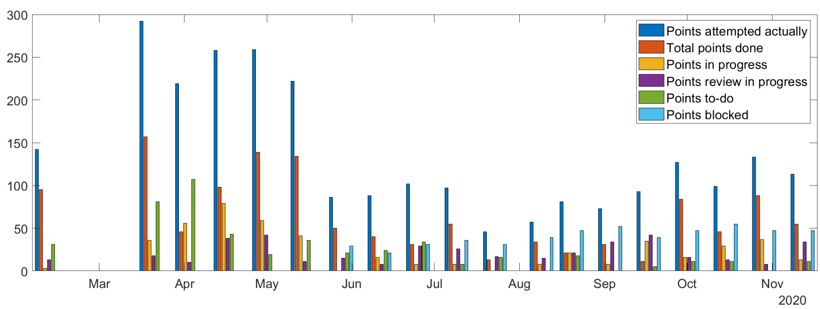


Figure 5.6: Full software sprint statistics barplot.

5.6 Paper F: An Agile Systems Engineering Analysis of a University-built CubeSat

Agile SE “refers to the adaptability and sustainment of adaptability [171]”, and is concerned with being agile in both the process applied to developing systems — and to the systems themselves. The agile method has perhaps gained the most popularity in software development through “Manifesto for Agile Software Development” and is increasingly applied in the hardware and other domains as well. The INCOSE Agile SE Working Group developed an “Agile Decision Guid-ance (ADG) Method” (described in Section 4.7.1) [241] which is used to analyze the HYPSO organization in order to identify areas where the project can benefit from agile approaches. Through a 12 week assessment addressing over 100 questions in a questionnaire, agile characteristics and challenges are identified. Within the product development space, increasing the visibility of changes is identified as an area to focus on to lower the impact that the support environment (e.g., machining facilities) has on the organization. For the solution space, the analysis shows high variability and dynamics of solutions, which increases the workload on the team.

5.7 Paper G: Addressing the Sustainable Development Goals with a System-of-Systems for Monitoring Arctic Coastal Regions

Developing systems for monitoring the Arctic coastal regions allows decision-makers to develop strategies for sustainable management of these resources. The vastness and challenging environment of these regions mean that it is not cost-effective to base the administration on a single technology for monitoring with the required spatial, spectral, and temporal resolutions. This paper looks at a MASSIVE from a System of Systems (SoS) perspective and describes how it can support the sustainable management of the Arctic coastal regions of Norway. The analysis method Stakeholders, Problem, Alternatives, Decision-making, Evaluation (SPADE) is applied [258]. Stakeholders range from government institutes and departments, end-users, technology providers, to regulatory bodies. The problem is that a single system cannot provide the information needed for monitoring coastal regions, and the communication infrastructure is not sufficient for arctic operations. Four use-cases are developed as a basis for discussion for oceanographic monitoring of the coastal regions, and MASSIVE capabilities are analyzed to evaluate how it may fulfill the stakeholder needs. The analysis looks at what CS need to participate and what they offer to satisfy the use-cases, and what decisions need to be made moving forward. The MASSIVE project as an SoS is summarized in Table 5.1, from Paper G. Reasons for viewing the MASSIVE project as an SoS were twofold:

1. The project team desires to avoid the failure to recognize and benefit from synergies between CS in the solution space such as coordinated ocean observations, and,
2. The number of CS and their communications are too complex to handle as a single system

Table 5.1: The MASSIVE project as a System-of-Systems according to Maier's five dimensions [9].

Dimension	Description of MASSIVE
Operational independence	Each of the CS are developed to operate independently and can reach decisions without the other elements to perform their own mission objectives.
Managerial independence	The CS are developed in different phases, and some have higher maturity than others because of this. As an example, the satellite system can be developed and perform independently as a sensor system without the presence of other parts of the multi-robot system.
Evolutionary development	Evolutionary development of the CS allows the SoS' capabilities to evolve with technological advancements, which in turn motivate new capabilities.
Emergent behavior	No single CS can monitor the coastal and Arctic regions with the timeliness and level of detail required without cooperating within the SoS.
Geographical distribution	The developing organizations are not co-located. Also, the CS only interact through information or data exchange and do not rely on physical interactions.

5.8 Paper H: MBSE Modeling of a SoS with a Small Satellite and Autonomous Surface Vessels for Persistent Coastal Monitoring

One of the use-cases described in Paper G is further explored in Paper H, applying MBSE to the SoS to explore possible architectures and capabilities, and identify technical and managerial considerations that should be made for the system design. The use-case of collecting high and low resolution information on algal blooms is analyzed. Three alternative scenarios with associated system architectures are suggested, and pros and cons of these are explored further in [35]. The process is supported using Arcadia Capella for the MBSE effort. In the paper, we discuss how modeling the SoS with different viewpoints enabled better identification and allocation of functions. For example, that the elaboration of exchange scenarios also identified missing system functions and the need for better coordination between the development efforts of the CS. This coordination entailed agreement on the data to be exchanged, documentation of the technical specification for the communication system, and analysis of the impact of the interface on the collective data budget for the SoS.

5.9 Paper I: Academics' Perception of Systems Engineering and Applied Research Projects

Project-based research has become dominant in funding and organizing research efforts, which can be seen in academia as well [49]. The motivation for conducting this analysis was to understand how a university executes projects concurrently with research, how academic staff view projects, and what opportunities exist for improving the system to support researchers in balancing the workload of performing in these different roles. The research is based on a qualitative case study, including a literature review and 18 semi-structured interviews lasting 45–60 minutes. We find that the opinion of integrating SE in academia was linked to the interviewees understanding of SE, and that SE practices were common sense and recognized as approaches for knowledge management. In addition, we see that academics differentiate between project and research management, that projects have clearer objectives and goals, while research management can happen on many time scales and needs, and the approach may need more tailoring. The interview analysis shows agreement on lack of clear guidelines for processes or methods for managing projects at the university. Moreover, there are different opinions on what constitutes good research management, and the results indicate that iterative approach and collaboration between technical and project management is positive for research projects. While SE provides a holistic overview and structure to a project, the interviewees were critical of applying too strict processes to research projects because it could hinder creativity and requires too much resources.

5.10 Paper J: Towards an Integrated Project Complexity Narrative – A Case Study of Academic Organizations

Building on paper I, the increased complexity of applied research projects makes it more challenging to deliver good research results. The research projects hinge on the capacity to manage interactions between people, organizations, technology, stakeholder politics and business interests in a cohesive and holistic manner. However, how to manage the complexity of research projects in academia has received little attention [259]. The research is based on the same interviews as in Paper I, and how the findings can inform the effort towards a complexity language [260–262] that can be used in academia, based on the following five dimensions of complexity: *structural*, *uncertainty*, *dynamics*, *pace* and *sociotechnical*. The interviews are analyzed to look for evidence of these five dimensions of complexity when speaking of projects in academia. We find evidence of structural complexity of systems, uncertainty of projects and the organization, and pacing of research projects and how this differs depending on the project and the goal. For example, how rapid prototyping can support faster development cycles, or how projects sometimes turn into programs. Regarding the sociotechnical complexity, interviewees differentiate between interpersonal, societal, and organizational complexity, and challenges with navigating these areas. Our findings indicate that any consistent differentiation between concepts of complicated and complex is lacking. Furthermore, when addressing characteristics of complexity informants focused on physical and logical systems. Such language challenges could arguably hold groups back from greater effectiveness in managing social-political risks in their work.

Chapter 6

Analysis and Discussion

Science and everyday life cannot and should not be separated.

Rosalind Franklin

In academia failure is recognized as a part of the learning process. While universities may want to have project success rates comparable to industry, they also must produce research results and prepare tomorrow's engineers in less time and with smaller budgets. To contribute with cutting-edge research involves greater risks which universities are willing to take because students learn much from experiencing failure or making mistakes.

The original objective of this PhD research is to build knowledge for the understanding of applied research projects in academia. A goal is to use this knowledge to improve the way academic organizations build systems for space, so-called "rapid systems engineering". The increased projectification of research in academia introduces the need for better project management and systems engineering activities in multidisciplinary projects. Moreover, research projects involve people, and managing them requires a sociotechnical approach. The research questions were addressed with different perspectives throughout the research period, and a summary of the main points follows.

In this chapter, I provide an analysis of the results concerning the research objectives and questions. I also analyze the research limitations in terms of the research design and methods applied throughout the PhD period. The chapter is structured as shown in Figure 6.1.

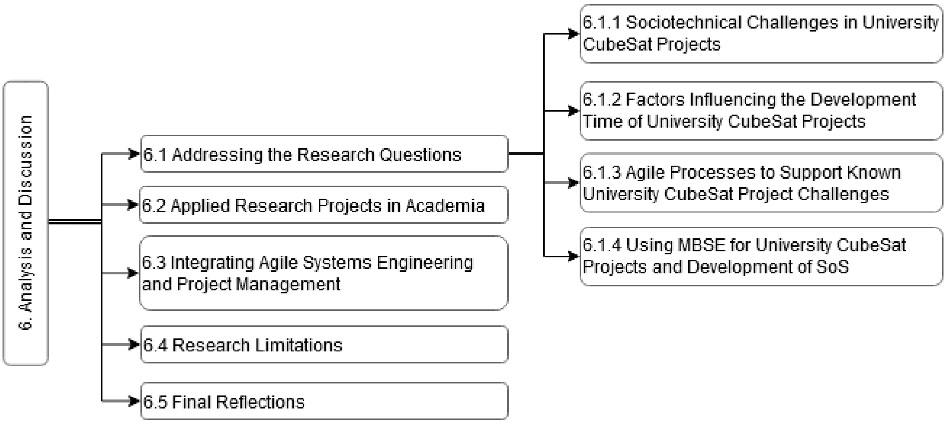


Figure 6.1: Outline of Chapter 6.

6.1 Addressing the Research Questions

6.1.1 Sociotechnical Challenges in University CubeSat Projects

RQ-0: What are known sociotechnical challenges in university CubeSat projects and how can they be addressed?

This question is first addressed in the literature review (see Section 3.2) and then by the longitudinal case study of HYPSO. University small satellite projects have gained traction since the CubeSat design specification release in the late 1990s, and their objectives range from being educational to delivering scientific data to in-orbit demonstrations of new technology. The main contributions towards RQ-0 can be found in Papers A, C, D and F.

The literature review highlighted the following challenges: (i) knowledge management; (ii) incorporating lessons learned systematically; (iii) schedule overruns and lack of funding; (iv) little testing at system-level; (v) lack of formal methods for risk and failure analysis; (vi) successful integration of CubeSat engineering tasks into the curriculum.

In Paper A, the knowledge management aspect is highlighted as a challenge for the HYPSO team. Since the team is distributed they do not communicate enough, only through weekly meetings, which in turn leads to rework because decisions are made without the whole team understanding or learning the rationale. This may partly be because the different disciplines are not aware of how their choices impact the overall system design, i.e., they lack a holistic view of the system. Furthermore, a lack of a formal methodology for risk and failure analysis makes it challenging for inexperienced team members to see the effect of design choices on the overall mission success. There is a need for processes to enable better knowledge sharing and building a common mental model of the system.

Paper C goes deeper into understanding the factors that contribute to the above challenges, by categorizing them into different factors, also contributing to RQ-1. Again, knowledge management and team knowledge are highlighted as recurring challenges for the HYPSO team. In addition, having a clear picture of the mission objectives and scope, the system architecture and requirements, could enable successful onboarding of new students. The planning and task prioritization is challenged by the volatility of external and internal facilities. Combing with a lack of transparency of the academic school year across a mix of disciplines can lead to schedule overrun.

In Paper F, the Agile Decision-Guidance Method (ADGM) is used to pinpoint factors of the HYPSO organization that can benefit from an agile approach. Also, the process reveals areas where the team experienced sociotechnical challenges:

- Goal misalignments between students wanting good grades and the HYPSO project manager wanting a successful mission.
- Balancing an information system and knowledge management.
- Requirements management and ownership.
- Developing a common understanding of the system.
- Interface management.
- Dependence on the internal and external support environment.

Paper D explores the challenges of managing a university CubeSat team during the COVID-19 pandemic. These experiences cannot easily be compared to literature because there have not been many similar situations in recent history. There have been "planned" virtual teams, and there is much literature on risk management. However, the lockdown presented a new situation for universities, and the opportunities that exist with digital infrastructure. First, technical infrastructure is key for collaboration and design work. While the HYPSO team had been moving towards a more digital infrastructure for some time, the lockdown made the need for a solution both immediate and urgent. Second, there must be a workflow associated with the infrastructure. For example, the HYPSO team used a GitHub workflow supported by daily stand-ups through a videoconferencing tool and an asynchronous chat platform. Third, since there was no lab access, test setups such as HIL that were integrated with the GitHub workflow, is needed for continuous system testing [263].

The sociotechnical challenges associated with the lockdown included motivation, diversity management, knowledge management, and understanding how risk management is affected by a crisis. For example, many of the team members went from seeing each other every day in the lab and meeting classmates on a regular basis, to not meeting anyone in person. Maintaining the HYPSO team culture and understanding the team members' diverse situations requires attention from both project management and technical management. While managing diversity was not a new sociotechnical challenge, it gained a new dimension in that for some students, the daily stand-ups and team interactions were the only interactions they experienced throughout the day. Arguably, the agile practices that had been introduced a year before, greatly decreased the negative effect on the planning and scheduling by the pandemic. For example, the team already had a culture of responding to changing academic calendars, and was therefore able to adjust to the changing circumstances the pandemic created.

The literature offers suggestions for how to address these sociotechnical challenges, such as having some staff members leading the university CubeSat projects continuously and thus maintaining culture and knowledge even with student turnover; short design cycles; mixing curricular and extra-curricular work; version-controlled repositories; more testing; applying processes and standards; and incorporating dependability work early in the design cycles. Specific methodologies such as agile Scrum, database-based mission design, and concurrent engineering practices have been suggested and applied for university CubeSat projects to varying degrees of success as described in Chapter 3.

6.1.2 Factors Influencing the Development Time of University CubeSat Systems

RQ-1: What factors influence the development time of university CubeSat systems?

To build a collaborative project culture and increase knowledge sharing, good onboarding processes and kick-off activities are important [104, 150]. The project manager should ensure that team members follow the same workflow to know what to expect from each other. The N2-matrix was useful to visualizing the interdependencies between people and how their deliverables impacted others in the team (shown in Appendix 8). This supported fruitful discussions and coordination between team members whose work depended on others. We found that it was important to build, test, and fail quickly to learn quickly. This agrees with much of the literature on building CubeSats in universities [22,24,26,28,51]. The iterative process of failing and learning is facilitated by a mixture of **parts supply chain**, **lab facilities**, **university facilities**, and **external facilities**. The students gained quicker learning by having enough parts to prototype with and test-to-destruction instead of just studying the same parts with analysis and datasheets. **Lab facilities** in this context mean having target hardware for embedded software development and verification, spare parts, basic tooling that students can use without specific training, and a place to work together. Throughout the case study, this has been critical for development, also supported in much of the literature [22, 28, 51].

In many universities, projects are classified as **flagships** (linked to national government programs) or **independent schools** [51]. The HYPSO project would be classified as **independent schools** and therefore should be compared to those, **Independent schools** do not have the support of government bodies or an established supply chain of test facilities. For example, if all the facilities are in-house, there is no need to coordinate with external facilities. However, this can result in higher maintenance costs and resources to support the in-house facilities. While in-house facilities could reduce the development time, it is not realistic that all universities should have a full suite of small satellite testing facilities.

While using COTS components and solutions may introduce challenges for achieving compliance with requirements for space systems, as the findings from the HYPSO case study show, they are still recommended for academic CubeSat projects. However, since datasheets and information may be lacking from the suppliers, there is an increased effort required for verifying that the COTS components will work in the integrated system. We found that testing to destruction, following best practice recommendations from other CubeSat projects, and applying standard interface protocols where possible, greatly improved the value of the COTS components. Even if the components are not performing as expected, the primary mission objectives may still be fulfilled by changing the system requirements or operational scenarios. This should be done in collaboration with the stakeholders.

The external factors, such as **external facilities** and **parts supply chain**, are challenging for university projects because they often lack the necessary business relationships to maintain an adequate supply level. This became even more evident during the COVID-19 lockdown, where the access to external (and internal) facilities was blocked. Moreover, universities have fewer resources, and cannot pay as much as a commercial client, which means external facilities may not prioritize academic customers. However, by supporting academic projects, the external facilities can garner goodwill and positive media coverage and may also support their recruitment efforts. Establishing relationships with a network of external facilities proved to be critical for developing the HYPSO-1 CubeSat. This helped knowledge building and provided physical facilities such as those needed for vibration testing or surface treatment.

Many CubeSat projects have experienced delays due to limited access to external facilities [22], which are out of the university project's control. The findings from Faure et al. [24] also recommend having suppliers nearby to minimize transport and logistic waste. Since students often work during weekends, external facility usage could be scheduled during off-peak hours. We suggest using an agile approach to managing external facilities so that the team can quickly turn around to e.g., do an interim vibration test, even though test plans are not finalized or the design is not ready. Early testing can shorten the development time by providing confidence (or proving failures) of mechanical design, which otherwise would not have been discovered until flight acceptance testing.

This study indicates that the successful onboarding of students is linked to how clearly the scope and objectives for the project were defined. Many team members highlighted the importance of having good interface documentation, which is recommended by ESA SE practices and in line with previous studies [24, 51]. Unsurprisingly, having good interface documentation, improves integration activities which can lead to discovery of bugs earlier, or designs to fail quicker. While having good **team knowledge** that cover all aspects of developing space subsystems is important, it is expected that the team's information system will "contain" most of the relevant knowledge by the end of the first mission. The literature concurs that teams launching more than one satellite have a much higher success rate and shorter development time [25, 51]. Similar results should be expected for the HYPSO team if they continue developing CubeSats, and especially for the aforementioned second HYPSO satellite.

The analysis shows that CubeSat project managers can use the readiness levels (TRL, IRL, and SRL) to keep track of their subsystems' maturity levels. These will inform the managers where to focus their activities and efforts. However, given the outlined limitations (see Section 4.7.4) regarding the non-valid arithmetic operations on TRL and IRL to achieve SRL [255], the calculated SRL should not be used to compare systems when making a decision. However, the exercise of finding the TRL, IRL and SRL are valuable because it will highlight to managers and engineers which subsystems have low TRL and IRL and warrant more attention. A limitation to using the readiness levels is that the assessment strongly depends on the people assessing, the context, and the organizational culture [249]. Thus, the maturity assessment should be done in a group, not by single engineers. However, the same components could be assessed differently in a different organization. Therefore, the assessment is mostly valid within the same organization and culture, and not recommended across organizations, except perhaps as an indication. Improving the semantics of the levels and standardization could help address this limitation. Furthermore, there is a need for a shared information system where there is a balance between push and pull information flow to ensure that relevant knowledge is available to team members. As the program at NTNU continues, the information system should allow for the reuse of knowledge.

6.1.3 Agile Processes to Support Known University CubeSat Project Challenges

RQ-2: To what extent do agile processes support known university CubeSat project challenges linked to knowledge management, system testing, project management, and team composition?

The agile management approach is grounded in enabling the ability to respond to changing circumstance and new discoveries. It also means empowering the whole project organization to participate in decision-making, as opposed to relying on the manager to decide the scope and team activities. There will still be a need for management activities, but the top-down hierarchical chain-of-command is replaced, thus reducing some of the asymmetry between power and influence over work.

From the HYPSO case study, it was found that the project exhibits many characteristics that makes it suitable for an agile development approach, as opposed to a stage-gate approach. When considering the boundary conditions from Paluch et al. [77], and characterize the HYPSO project, it is clear that the agile approach is appropriate, shown in Table 6.1.

Moreover, the ADGM analysis (Paper F) suggested factors for improvement such as increasing the visibility of changes for the support environment, also identified as one of the factors highly influencing the development time in (Paper C, E). Since the solution architecture has been, and is evolving, the adoption of Agile SE system architecture principles around reusability, reconfigurability, and scalability [171] can be used to proactively move to a more agile architecture in future HYPSO projects. The software architecture chosen (Service-Oriented Architecture) for HYPSO [263] is suitable for a “living scientific software product”, and works well with an agile workflow.

The tailored Scrum methodology, where team members could deliver both product increments and thesis iterations, worked well for the HYPSO team. While we did not see any sprints where all points were completed, similar to Garzaniti et al. [83], this does not mean that the sprints failed or that the approach is not suitable. The approach has been tested on two different groups of people: the 2019-2020 team and the 2020-2021 team, with the same Scrum leader. One of the challenges with closing software related issues was identified as code reviews, which can be challenging when students work on modular subsystems and do not have the knowledge to review each other’s code. The tasks then is left to the group leaders, which do not have time to review everything each sprint. The team members also expressed that having sprints helped them organize their work. Tailoring the Scrum methodology to include project work and thesis work is recommended for helping team members balance their total workload. The case study showed that it took some time for the teams to adjust to working in sprints and learning how to score issues [83].

The HYPSO project chose to have daily stand-ups, as opposed to weekly [51]. This frequency proved very useful during the COVID-19 pandemic lockdown. It provided both a daily human contact point in stressful times, and allowed group leaders to catch any unforeseen impacts from lockdown or other national measures faster. Furthermore, daily check-ins on progress can reduce time wasted, a recommended lean practice from Cho and Mazui [22]. If team members are having challenges with an issue, they can reach out to get assistance during the stand-ups. The data did not show correlation between attending stand-up meetings and thesis performance, but there were indications that project tasks were executed faster or with higher quality with high attendance. If they have completed the work and need more tasks, they can acquire these at the stand-ups. Both mechanisms reduce the wasted time.

My results support additional recommendations from Berthoud et al. [51], that version-controlled repositories and regular face-to-face interactions are essential. Furthermore, that issue tracking systems such as GitHub are useful and can help both as a source of information and to understand project’s evolution (e.g., lessons learned). My research indicates that project documentation such as design and test reports are accessed more often than previous theses as sources of information. This means that in addition to writing their theses, students need to write good project documentation. The HYPSO team encourages this in two ways. Firstly, whenever someone finished an issue in Scrum, they have to document it either through code or formal project documentation. Secondly, by conducting formal design reviews at the end of each semester the project documentation they have written is collected into design reports for the formal design review. Furthermore, when following the GitHub workflow, people have to review each other’s code, which means it needs to be documented well. This knowledge repository and workflow encourage people to write good project documentation. Many students also include project documentation as part of their results in appendices of their formal theses.

6.1.4 Using Model-Based Systems Engineering for University CubeSat Projects and Development of System-of-Systems

RQ-3: To what extent can Model-Based Systems Engineering support university CubeSat projects and the development of System-of-Systems?

The HYPSO project adopted MBSE practices for different SE activities, including: requirements management and traceability to components and verification activities, dependability analysis, and for architectural considerations of the MASSIVE SoS. The publications contributing to this research question are Papers B and H. Paper G establishes the basis for Paper H.

Paper B reports the results of implementing Failure Mode, Effects, and Criticality Analysis (FMECA) in MBSE. There are inherent limitations when applying FMECA to COTS components that do not have enough failure data to be quantitative, but this is not subject of discussion here. Incorporating the dependability analysis into the system design makes progress towards addressing the gap described in [264]. This approach to dependability analysis is becoming more evident in the literature [158, 265–268], and my findings support these research findings. Traceability between MBSE elements (component/function – failure mode – failure cause – failure reduction) helps the team members understand how the design is affected as a whole, and how failure modes can be addressed in several parts of the system with different failure reduction activities. This can be used to evaluate efforts needed and prioritize activities. Failure reduction measures can also be traced to new functions or components in the design. These updates and traceability contribute to knowledge management and understanding of the system design. The case presented in Paper B is limited to one CubeSat payload component, which yielded positive results. The results from the FMECA workshops were made available to team members in form of worksheets and a bachelor thesis [269], and then the results were analyzed using MBSE. It can be argued that taking a structured approach to analyzing the worksheets could have given similar results. To provide more value, the MBSE dependability analysis could be expanded to include use cases, or a functional Fault Tree Analysis (FTA).

Small satellite systems and the associated ground systems needed to operate them can be considered an SoS. When viewing MASSIVE as a whole, this SoS becomes more complicated, because this involves multiple autonomous assets, weather services, and others, which can choose to join or leave the SoS [270]. Therefore, system integration will occur on multiple levels. First, on a spacecraft level integrating NTNU subsystems with the spacecraft supplier subsystems. Second, integrating the spacecraft with the ground segment and ensuring that the operator can monitor and control the spacecraft to deliver data to the end-user. Third, the spacecraft and operator SoS need to be integrated with the overall MASSIVE SoS.

The findings reported in this thesis indicate that taking an SoS approach to viewing the MASSIVE project supported the design process of the different CS and identified missing functionality and interfaces by establishing an overall SoS architecture and concept of operations. We found that the MASSIVE SoS can be classified as something between a *collaborative* or an *acknowledged* SoS. The CS have independent management and can evolve independently but must collaborate to fulfill the MASSIVE mission objectives [75].

The results from Papers G and H show that having an SoS perspective allows us to:
Identify the stakeholders and their influence over the SoS. Particular attention should be paid to how stakeholders could influence the integration activities and whether CS stakeholders can inadvertently hinder the development of the SoS. For MASSIVE, it was important to set up regular meetings with CS developers, and coordinate the integration schedule with the stakeholders for each CS. For example, the AutoNaut would go on missions to fulfill scientific objectives. If we wanted HYPSO and other actors to support that, we needed to set up interfaces for that. Or, to postpone integration testing until after these missions.
Develop SoS architecture. An operational, system, and logical analysis was performed in Capella using the Arcadia method. The operational and system analyses showed how different exchange scenarios could provide the required data to end-users but with different performance characteristics. The logical analysis enabled the allocation of functions between CS and highlighted missing functionality or interfaces.

The results are strongly aligned with the recommendations from Dahmann et al. [121], alleviating some of the pain points when integrating an SoS [75]. The use of MBSE to support the SoS analysis allows for capability definition and architecture development [63]. It does not sufficiently capture important management considerations that should be made when integrating CS into an SoS. This should be supported by other methods, for example, systemigrams, which is suited to model sociotechnical and complex system interactions. Furthermore, the modeling effort requires resources in terms of training and actual implementation. Modeling efforts should follow best practices and define the goals and scope of modeling first.

While not fully incorporated for HYPSO-1, a system model with traceability to user scenarios, requirements, functional flow diagrams, interface descriptions, etc., can be reused to ease development in the early phases for HYPSO-2. Example of diagrams:

- Functional flow block diagram is shown in Figure B.7.
- Breakout board failure modes and causes in Figure B.14.
- Failure reduction in Figure B.15.
- Mission objectives for the HYPSO-1 mission in Figure B.17.
- Mission requirements breakdown and associated verification requirements and component specifications in Figure B.18.
- Verification event traceability diagram is shown in Figure B.20.

The system model can be based on the CSRM developed by INCOSE Space Systems Working Group [70]. Recent studies [69, 188, 213, 214] have shown utility in using the CSRM. The CSRM is recommended for project organizations such as HYPSO, where students have a mix of backgrounds and limited time available for spacecraft development. MBSE supported by reference model can provide increased rigor to SE activities [66]. Moreover, the system model can include verification and validation templates and be linked to reliability models. Thus, it would be possible to keep information from one project to the next, informing future teams on typical failure modes, verification activities, and where they would occur in the lifecycle. The system model could include mitigation measures to help new designers. The model could also support requirements with traceability to components and functions, which increases the team ownership to the requirements. An improved solution architecture should make the requirements management simpler, and the team could also investigate principles for lean requirements management [271] to minimize the efforts needed.

Using MBSE over e.g., flowcharts and drawings, is beneficial because the process based on formal modeling language with semantics, which flowcharts and drawings do not [272]. Building on the formal modeling language and semantics, most MBSE tools offer validation checks and model consistency checks. There is still a risk of modeling the wrong system and functionality, however, internal validation (such as for Paper H, where we worked on the models in several consecutive meetings to make sure it represented the system) coupled with model consistency checks can mitigate this risk. The system model and elements can be shown in different viewpoints, depending on the need of the designers/stakeholders accessing the model. This supports the alignment of interests and understanding, one of the identified challenges for university CubeSat projects. It should be done in open source tools to keep costs low in university projects, and include guides for usage and, be linked to the overall lifecycle development workflow and planning. Learning MBSE in a university project can be beneficial for engineers, or even just being exposed to MBSE. It could lower the barriers to usage when these engineering students graduate and start working in a company.

Table 6.1: Comparison of the linear, agile, and research development environment. The linear and agile development columns are from Paluch et al. [77], while the *HYPSO project* column is based on the findings of the case study.

	From Paluch et al. [77]		HYPSO project
Characteristics	**Linear development**	**Agile development**	**Applied research**
Solution space	Solution space defined	Solution space undefined	Solution space undefined Theoretical grounding
Customer	Stable and known customer preferences Limited customer willingness to interact Customer in need of fully specified product	Changing and/or unknown customer preferences High customer willingness to interact Customer open to engage with interim products	Changing interests from the research community High willingness to review Research community open to contribute to products
Task	Low task modularity	High task modularity	High individual research modularity Managing researchers and research projects Applying for research funding Explore the topic in depth and develop knowledge
Goal	Well-defined and agreed-upon goals	Open and agreed upon goals	Open goals, not well-defined Creating knowledge for a better world Answering research questions
Process	Well-defined and standardized process	Adaptive process models Continuous integration, test-driven	Weakly defined, but highly adaptive process Highly learning-focused process Known and declared methods in some fields
Organizational	Low tolerance for interim failure Strong need for managerial control	High tolerance for interim failure Weak need for managerial control Shared ownership	Weak need for managerial control High tolerance for interim failure High acceptance for individuality

6.2 Applied Research Projects in Academia

The results of the PhD project do not clearly show how an engineering project could ensure the fulfillment of academic research goals, but rather, how academics view research and engineering. There is an agreement that most of the projects related to space systems today require technology to push the research boundaries, build equipment to deliver research data, and research options to build better equipment to reach new insights. The findings are consistent with Bentley et al. [48] and suggest that the "projectification" of academic research projects means that universities could benefit from a more systematic approach to managing. Some of the findings suggest there would be resistance to incorporating PM and SE methods and heuristics because academics fear that these can restrict their creativity, which they consider essential to the research process. These results correspond with the findings in Malik et al. [147].

In addition, the study indicates that some academics are frustrated with the lack of planning of engineering resources from the academic department, i.e., that it is difficult to know when their research project could get support. Closer attention to which types of research projects need this additional support would enable departments to prioritize resources, and researchers to be assured of this support so that they can focus their time on research. Papers I and J are based on the 18 semi-structured interviews on how academics view SE and PM, and how they differentiate and talk about projects and research. While these findings are based on a key informant sampling method and inferences cannot be generalized, it raises questions for how academics perceive and manage applied research projects.

6.3 Integrating Agile Systems Engineering and Project Management

It is essential to build team culture and foster collaboration in any project since so many of the challenges and development activities are multidisciplinary. The HYPSO project is an excellent example of a multidisciplinary project, and the university context added additional challenges. The application of SE and PM activities and methods was helpful in managing both the technical and sociotechnical aspects of the project. The COVID-19 pandemic lockdown, which introduced significant and ongoing disruption from March 2020, changed the dynamics and context for the HYPSO project. In a changing environment, which impacts people and suppliers, SE and PM activities are important for ensuring project continuation. Furthermore, the team was empowered in making decisions, as opposed to relying on the project manager to decide the scope and team activities. There is a potential for people to make mistakes when there is a transition from nominal to emergency state, and information can be lost. It is during these transitions that SE and PM are most needed.

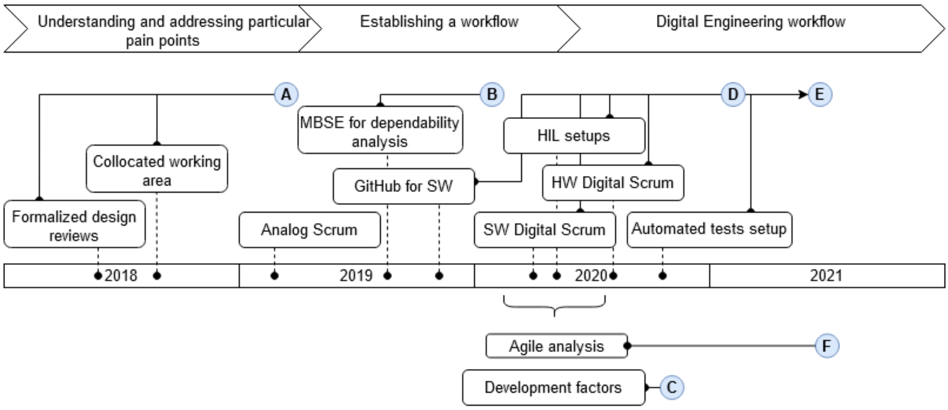


Figure 6.2: The research process and publications related to the HYPSO case study. The first phase focused on addressing individual issues and understanding the challenges and context. The second phase looked at integrating improvements into a more holistic workflow. The third phase and moving forward focuses on moving towards an integrated Digital Engineering approach.

The students lack experience in planning their work and estimating the time a task takes or what impact the tasks have on their overall workload. Lack of planning experience coupled with limited time and resources [111] meant that following a traditional SE and PM methodology as described in the ECSS and NASA standards is not suitable for the HYPSO project. Therefore, an agile approach to SE and PM was needed, where which activities to include and leave out could be tailored.

A proposed methodology incorporating the lessons learned from this research is given in Figure 6.3. The sprint planning needs to take the academic calendar and external facilities and support functions into account. Deliverables can be thesis work, hardware and software modules, verification and validation activities, and other parts of the system model. Not shown in the figure are the design reviews that happen at the end of each semester. The design reviews are helpful for students and the stakeholders, because they are milestones in which the current design is agreed upon. However, if new discoveries which could increase the value to the stakeholders are made, the plan needs adjustment.

The data collection for the HYPSO case study happened at the same time as the studies published by [83, 240] and [78, 239] (from Table 4.6.) The choices of applying agile with Scrum for software and hardware development, using a FlatSat approach and multiple development models/prototypes in these projects occurred in parallel and independently of each other. In the HYPSO project, the thesis work and academic calendar have also been included as part of the Scrum work. According to the information available from the other cases, this is a difference. The HYPSO CubeSat and the Iris CubeSat have similar characteristics in terms of design philosophy (e.g., use of COTS and in-house subsystems for the satellite bus, and developing the payload in-house.) Furthermore, these projects have similar approaches for SE and PM, applying agile to the entirety of the project.

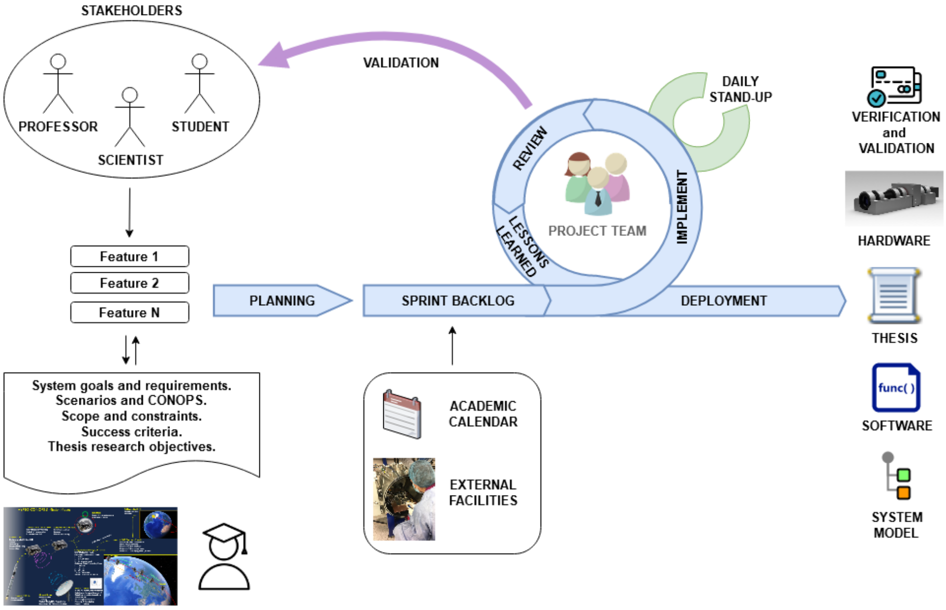


Figure 6.3: Pragmatic methodology for integrating agile Systems Engineering and Project Management for Small Satellites in academia.

In the HYPSO project, Digital Engineering practices are actively applied as shown in Figure 6.4. To a certain degree, this is the case for the Iris CubeSat as well [239], and we can assume for the unnamed CubeSat reported in [240]. However, it is not clear if these projects have taken an integrated and conscious approach as the HYPSO project is [16]. The Digital Engineering approach enables better integration between SE and PM because of increased transparency between technical work and project work, and can contribute to a better information system.

A waterfall planning approach was tried for some months before the agile approach was chosen. However, as the team grew and more students joined, it was difficult to take different coursework demands into account, and the plan was not reactive enough to the changes. Furthermore, the different theses tasks influenced the development of the system and functions. Because of my, and some of the other team members, familiarity with the agile approach and its values [1], this approach was chosen for the HYPSO project. The agile approach also lent itself well to the GitHub workflow introduced in the software team, and the HYPSO team consists of mostly software-related members (on-board processing, operations, data distribution, etc.)

Whether an agile approach such as Scrum is appropriate for university CubeSat teams can be discussed, and whether it is efficient in all phases of development. The principle of design iterations can work well in stage-gate approaches too, if the iterations are planned. Hardware development still relies on certain long-lead items, and these design decisions must be made early to meet

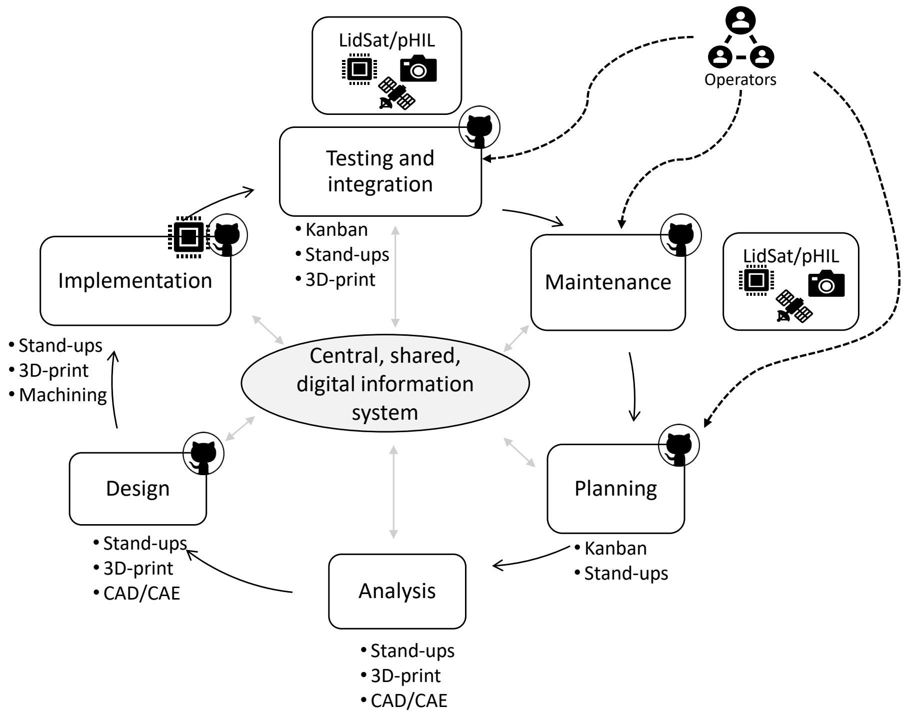


Figure 6.4: Digital Engineering system lifecycle.

an overall timeline. Furthermore, the assumption that the system does not have any security- or safety-critical functions (except to not be harmful to other payloads on the launch vehicle), may not be the case for future satellites. In that case, more hybrid models of agile stage-gate development should be explored, to find the right balance between risk and responsiveness.

6.4 Research Limitations

The central part of this research is based on a longitudinal case study of the HYPSO project, with Participatory Action Research as the method. I have not attempted to summarize the results and findings from the case study into a single sentence or hypothesis because explaining sociotechnical behavior is complex, and the context is important. I have used various literature to study the phenomena observed and borrowed from both SE and PM to analyze the results. By publishing papers in a variety of peer-reviewed fora with different viewpoints and findings, my intention was to give external validity to the conclusions at the same time offering a broad perspective of the case study. However, each reader's background will influence their interpretations of my publications, making each individual's learning experience different. Some readers will find external validity of certain aspects in the findings, while others may object to the conclusions.

In each of the publications and associated findings, I have described how they pertain to the current state-of-the-art, and in most cases, I found that there is an agreement with current practices and findings. There will be sources of literature that I have missed, which could have illuminated the results further.

Modeling techniques have been employed throughout this PhD research. However, models will only provide representations of the real-life system, according to viewpoints and interpretations at the time. Representing the human systems within the system context remains a challenge, although attempts at "snapshots" have been made (for example, in the systemigrams). The modeling techniques (systemigram, Arcadia method, GENESYS/Core, N2-diagram) chosen do not accurately represent the variability of human systems and how the human systems evolve as they learn.

The measures for addressing validity and reliability through triangulation is given in Table 6.2. Triangulation is defined as "the use of more than one method or source of data in the study of social phenomenon so that findings may be cross-checked [273]."

Table 6.2: Measures for Addressing Validity and Reliability

Method	Approach	Implementation
Data source	Different projects groups in evolving project phases	Longitudinal case study with different teams, Literature review, Case study on PM/SE views
Data type	Quantitative and qualitative data collection	Surveys (quantitative), AR and interviews (qualitative)
Researcher	Involving other researchers in data analysis and writing	Co-authors in publications, Review of publications by supervisors, Multiple interviewers present, Researcher triangulation
Method	Using different methods for data collection	Document review, Case study, AR, Interviews, Questionnaire

Case Study Case studies are in general limited in their external validity (the transferability) of findings because the findings are strongly associated with the context. The case study was that of a longitudinal study of the satellite project through many phases, and there can be value to readers to gain from the knowledge provided, even in its limited context. The subsequent HYPSO satellite projects (such as HYPSO-2), or other CubeSat projects at NTNU can use the findings and inferences to improve their project organization and processes. Most social systems exhibit features that cannot be generalized, because the systems studied are complex, not because the case study itself is limiting. It was only one longitudinal case study, so difficult to validate the findings, however abductive inferences can be discussed with team members to increase validation of findings. For internal validity, rival explanations such as "the null hypothesis", i.e., that the changes observed are by chance only and not because of the interventions such as agile practices, exist. Discussion of rival explanations with case study participants lowers this possibility.

For replication case studies, the objective is to confirm or disconfirm a claim, sometimes called hypothesis testing. The hypothesis testing can be done in a confirming or disconfirming manner. This process is highly influenced by confirmation bias. As the project manager of the CubeSat project studied, I am of course motivated by and interested in the outcome of the project, which may influence the treatment of evidence. Therefore, it is difficult to construct the case study to seek out disconfirming evidence.

It is not possible to avoid confirmation bias completely, or any of the cognitive biases. The biases influence the research paradigm and the selection of evidence. However, the research objectives and questions have been grounded in the literature review, and the methods chosen are in line with prevailing methodology for these types of studies. The field notes can contribute towards confirmation bias when focusing on what I as the researcher interpret as "important events". On the other hand, they serve as an evidence that can be used to disconfirm claims when reviewed, and it reduces the case of just recalling data that supports my biases. To mitigate the confirmation bias I have discussed the findings and data with several researchers (both researchers at the department, and also researchers at conferences) and considered alternative explanations. For example, whether the choice of a Digital Engineering methodology improved the development or if it is because of individuals in the team and culture. The longitudinal study and collection of data reduces the effect of individuals on the performance, although the core team are consistent. However, it was not possible within the PhD research period to find a case study where the core team changed. A further improvement would be to develop a case study with a different population to test some of the findings. For example, with different cultures or backgrounds, or at universities with longer experience.

Interviews and Questionnaires There are several limitations with using interviews, such as sampling bias, interviewer bias, and interviewee bias. A challenge in this research is that the interviewer could be considered an expert in the field compared to the interviewees. This can assist in carrying an informed conversation, but it can make the interviewees insecure or want to overperform if they feel like the interviewer is testing them [230]. Furthermore, the sample size of the academic environment studied is small, only two organizations in one country. For Paper I and J, a key informant method was applied, which is helpful in an exploratory study. A clear limitation of the study for Paper I and J was that only two organizations were studied, and the application of using interviews as the research method, and the key informant sampling method cannot support generalizations. Two researchers were present at most of the interviews to reduce some of the interviewer bias that can be present.

Action Research There are inherent challenges with Action Research since the researcher is actively participating in the case study. Norris [274] provides a list of common Action Research biases, namely, *reactivity of the researcher*, *selection biases*, *affinity of the researcher*, *ability of the researcher*, *value preferences of the researcher*, and *personal qualities of the researcher*.

For example, themes that would not gain much attention without the researcher may be over-emphasized, or topics may gain more support than without the researcher (confirmation bias). Furthermore, because I also was the project manager, power imbalance may also have influenced the practice changes. Throughout the participation in the HYPSO project, I have asked for feedback from participants on how actions are received or if things work better or not after actions were implemented. Further, we have had several lessons learned sessions so that team members could suggest themes I could have overlooked, or insights into themes that I did not consider. In addition, all published material has been reviewed by other HYPSO team members so that they could offer their comments and interpretations on events described in the publications.

6.5 Final Reflections

Was the choice of the HYPSO case study the correct one for generating knowledge to support the understanding of applied research projects in academia relating to space systems?
The understanding of applied research projects in academia is not mature, but there is a slowly growing body of knowledge on university CubeSat projects. Therefore, the case study approach is appropriate in the sense that there is not much statistical data available. However, the choice of a longitudinal case study, following a project from its inception, through COVID-19 lockdown, allows for a more in-depth study of a university CubeSat project, which was the objective of the research.

Were Action Research and interviews appropriate methods for studying the phenomena?
Throughout my time as a researcher, I acted as the project manager for HYPSO-1, and am continuing as project manager for HYPSO-2. Action Research as a method of for understanding applied research projects is appropriate, because it immerses the researcher in the case and gives first-hand experience of challenges and positives in this type of projects. In the beginning, I wanted to study more cases, but there were not many available for such in-depth study, nor did I have the time to get to know more than one project in-depth. Moreover, the bigger picture of MASSIVE and the goal to build organizational capabilities at NTNU for delivering CubeSats on a regular basis, made it an attractive case to study as a participating researcher. Interviews complement the Action Research by providing more viewpoints to the studied phenomena, and increasing the understanding of the project. However, both Action Research and interviews are biased by reflexivity, when the observed and interviewees only show and say what the researcher wants.

The research design evolved from attempting to find a methodology for developing small satellites in universities, to suggesting an overall approach for developing small satellites, through applying agile principles. The main reason for shifting was the growing understanding during the case study that I would not be able to find one method to fit every setting. Also, the project organization would benefit more from having a set of tools and methods to choose from, with suggestions for when to use what. I also went from believing that schedule was a driving parameter for academic projects to understanding that building an environment for continuous learning and research was much more important and that establishing mechanisms to learn from mistakes and allowing mistakes were key factors for building this environment.

Chapter 7

Conclusion and Further Work

Humans are allergic to change. They love to say, 'We've always done it this way.' I try to fight that. That's why I have a clock on my wall that runs counterclockwise.

Grace Hopper

This chapter presents the conclusions of the research performed during the PhD. The first section addresses the knowledge gaps and applied research projects. The second section provides a perspective on the contributions of this thesis. The final section suggests areas for future research.

7.1 Addressing the Knowledge Gaps

This thesis is a contribution to the discourse on how we manage our projects and resources, and what this means for the roles of Systems Engineers and Project Managers, especially in academia. The literature on university CubeSat challenges and lessons learned, teamwork challenges, interoperability and integration of constituent systems (CS) into an SoS all raise suggestions for ways to improve the sociotechnical challenges identified in the literature review and experienced in the HYPSO project as a replication case study. The research questions were developed to address the knowledge gaps for academic organization's capabilities for delivering and integrating space systems. Also, to understand how academics perceive SE and PM, especially ways the integration of these activities can support the fulfillment of academic research goals in a university context that is increasingly participating in multidisciplinary applied research project.

This section addresses the research gaps shown in Table 3.1:

Research gap 1:

> *There is a lack of awareness of the value of frontloading for projects and tailored methods for project planning for university CubeSat projects.*

In a university there is high turnover of students who follow the school year and the project teams face the challenge of balancing coursework and project work. This requires close coordination between SE and PM to manage knowledge sharing and transfer. A tailored agile PM process allows for transparency by the inclusion of coursework tasks and project tasks, which allows the SE and PM to make decisions to prioritize the critical tasks on the system development timeline (Paper F, E). An integrated, digital workflow based on agile principles is suggested as a framework for managing university CubeSat projects. The workflow incorporates increased system testing and iteration of design concurrently with building an information system through issue tracking and daily stand-ups to lower barriers to communication and increase collaboration. This research demonstrates how digital engineering coupled with an integrated application of SE and agile PM is well-suited to developing satellites in an academic context. This was demonstrated through the longitudinal case study of a periodically changing project team from multiple departments developing a CubeSat.

Research gap 2:

> *A pragmatic, conscious, integrated approach to testing in university CubeSat teams is missing. University CubeSat teams are either unaware of the need for tests or often lack resources to follow traditional standards for verification and validation. Better onboarding information is needed to provide the necessary instructions.*

The empirical findings from the HYPSO case study show how the use of HIL setups and early verification can contribute to shortening the system integration time (Paper E), and uncover system-level issues early [263]. The HIL setups can be an integrated part of the digital workflow to encourage continuous integration. When working with suppliers, setups using distributed FlatSats (such as in the case of HYPSO) can lower some of the costs of having to purchase all subsystems. Moreover, this approach encourages early integration testing and interface alignment with suppliers. Using COTS components can shorten the subsystem development time and lower purchase costs, but the integration of these can be challenging due to lack of transparent information in datasheets [46]. For space systems that use COTS elements, team members can build their understanding of the components and system faster by testing to destruction or doing early integration. Early integration and operational testing can reveal errors and failure modes impacting the system design with low cost. This shortens potential rework time and reduces the need for costly design changes late in the development cycle (Paper E).

Research gap 3:

> *An approach for managing knowledge and the availability of an up-to-date information system is lacking. Methods for improving the project planning methods to adjust for resource shortages, and understanding what influences the development time are needed.*

This thesis contributes with empirical findings from the HYPSO case study of sociotechnical challenges and what can influence the development time of a university CubeSat project. The case study calls attention to building a common mental model of the system, maintaining a vision, and a clear understanding of mission objectives and requirements as important factors affecting the development time. A shared file repository or system model with access to the updated project documentation or knowledge of who-does-what in an organization are important because development is happening concurrently, and developers need to know where to find current information and who to ask for clarifications (Paper B, D, E). Written and verbal communication between team members support team building and culture building, and lower barriers to sharing information. This can be supported by e.g., working in the same office area, attending stand-up meetings, sprint reviews and retrospectives, and design reviews. It is important to define the responsible roles and persons, and that the impact of decisions can be traced from function/element level to system and mission level. Having formal design review milestones also contribute to building a common mental model, because it makes clear what has been decided so far and there is a shared understanding of design (Paper B, E). This research contributes with empirical findings of ways digital platforms for design reviews, that can trace feedback from documentation to actions, can be used for project and product management. The empirical findings also highlight working with the external facilities and supply chain, and the internal support environment as factors that influence the development time. University projects may have little procurement power and resources to mitigate these factors, and being more responsive in working with external partners could improve the situation.

Research gap 4:

> *Additional research is needed for defining a pragmatic approach to work with vision, mission, requirements, and project scope in a team setting.*

In university CubeSat teams, the students join the team for a semester or two, and normally do not have prior experience working with space missions. This requires a pragmatic approach to onboarding and knowledge sharing, and to introduce the new team members to the mission and requirements so that they can contribute with value-adding functionality and deliverables to the project quickly. Teambuilding activities, frequent stand-ups, in-house workshops, etc. contribute to anchoring the team's vision and project mission (Paper A, D, E). For working with mission and requirements, this thesis shows that using MBSE can be used directed and pragmatically to address these pain points. The CSRM MBSE system model was useful to support mission development because it helps teams structure their development effort (Paper B, H). Systematic trade studies, verification and validation activities, and brainstorming activities enable better decision-making (Paper B, E). Using an SoS viewpoint supports the design process of individual CS because it enables the identification of the SoS stakeholders and their influence and development of functional and logical architecture (Paper G, H). The application of MBSE as a method for contributing to a SoS enables capability definition, functional allocation, scenario development, and identifying interface considerations (Paper G, H).

Research gap 5:

> *Additional practices are needed to define ways to integrate the reliability and risk management activities into the design process so that they are treated as a valuable design input and can be traced.*

This thesis demonstrates a method for integrating dependability engineering with the system design by incorporating FMECA in a MBSE tool. Establishing a system model that allows for reuse of components, e.g., in MBSE tools, can shorten the integration time when developing multiple systems (such as HYPSO-1 and HYPSO-2) that build on each other (Paper B, E).

Applied Research Projects
There is an increased projectification of academic research projects that involve several faculties and departments. These projects are often complex and highly interdisciplinary. The integration of PM and SE activities is helpful to manage complex projects, but there is little application or knowledge of such activities in academic research projects (Paper E, I, J). The results show that researchers distinguish between engineering and research by the tasks and goals. Moreover, that researchers are concerned that SE and PM can restrict the creative process of research. The strategic application of SE and PM practices to research projects is primarily dependent on the individual researcher. Researchers receive little training on how to apply SE and PM practices to their projects. Using the five complexity dimensions: structural, uncertainty, dynamics, pace, and socio-political, can support multiscale complexity evaluations in academic organizations by providing an ontology and shared understanding of how complexity can be characterized (Paper J).

7.2 Research Contributions

This thesis contributes to existing theories on applied research projects in academia, and, more specifically, managing university CubeSat projects. The theoretical contribution of this thesis is that agile Systems Engineering and Project Management practices are appropriate for managing university CubeSat projects, a specific case of applied research projects. This thesis extends existing literature and theory on the integration of SE and PM, and provides a practical contribution with a suggested framework for an integrated digital workflow, based on empirical findings from a longitudinal case study through several system lifecycle phases.

Although this has not explicitly been discussed earlier in this thesis, the synergy between academia and industry is an important part of academic research. The research conducted in this thesis sort of epitomizes this synergy, where I have an industrial background and experiences, and return to academia. The knowledge gained during my industrial experience feeds into the choices and actions for the academic research, and the management of an academic project. The subsequent knowledge gained from the academic research feeds back into industry, through meeting points like conferences, seminars, and other events. One confirmation of the value of the synergy between industry and academia is built on the empirical performance of the HYPSO team, which have responded to the changes and adopted the ones that worked well for their goals. This shows that we can take learning from industry back to academia, cycle the learning and return new students and new research results back to industry.

7.3 Further Work

The research reported in this thesis offers many potential future avenues for research.

Small satellite development: For small satellite development, there is a need for developing better information systems to reduce some of the overhead associated with documentation. Digital engineering and MBSE offer a promising way forward, but additional empirical study of this in a university context is needed. In addition, further empirical work which studies the application of agile SE and PM in an integrated manner with a defined workflow could provide new insights and recommendations. Most lessons learned information are shared in conferences or non-peer reviewed workshops. This makes it harder to learn from them when starting a new project, because they are not readily available for new players. More sharing of these kinds of case studies in peer-reviewed journals could greatly increase our understanding of small satellite development in academia, and also in industry. It would also be interesting to investigate how vertical integration of the verification and validation services can improve the development time for integrating small satellites and how this may affect the overall mission success.

System-of-Systems modeling: Future research in SoS modeling includes studying how emergence, resilience, and human systems in an SoS interact and contribute to the overall SoS mission. For monitoring coastal regions using a concert of collaborating autonomous systems with complementary sensors, this could be coupled to real-world missions. Future research could also look at how the SoS model can include aspects of digital twinning to enable better prediction of behavior and allocation of functions to constituent systems.

Synergies Between Academia and Industry: Research projects that receive funding from a university have goals for publications, for graduating PhDs, for dissemination of results, and delivering impacts. They often include a predefined schedule and limited resources, and it is not as easy as adopting standards that the industry uses for the execution of the research projects. Industrial practitioners are tackling the increased complexities of systems and projects by adopting standards and introducing new methodologies and practices. While some environments have good connections between industry and academia, this is not always the case. Future research could look at how the students from e.g., project based learning environments or applied projects such as CubeSat projects fare in industry, and explore the transfer of knowledge from these case studies into practices in the industry. Discourse on explorative projects and the role of system thinking in the context of academia-industry collaboration is a promising agenda and venue for further evolving multidisciplinary discussions on complexity, both in systems and projects.

List of References

[1] K. Beck, M. Beedle, A. v. Bennekum, A. Cockburn, W. Cunningham, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, “Manifesto for agile software development,” 2001. Accessed 2020-10-12, https://agilemanifesto.org/.

[2] ISO Central Secretary, “ISO 21839:2019(E) Systems and software engineering - System of systems (SoS) considerations in life cycle stages of a system,” standard, International Organization for Standardization, Geneva, Switzerland, 2019.

[3] California Polytechnic State University, “Cubesat design specification rev. 13,” 2018.

[4] P. Checkland, *Systems Thinking, Systems Practice*. Chichester, UK: John Wiley and Sons Ltd, 1991.

[5] Project Management Institute, *A Guide to the Project Management Body of Knowledge*. PA, USA: Project Management Institute, 5 ed., 2013.

[6] M. N. Sweeting, “Modern small satellites-changing the economics of space,” *Proceedings of the IEEE*, vol. 106, no. 3, pp. 343–361, 2018. https://doi.org/10.1109/JPROC.2018.2806218.

[7] OECD Statistics Directorate, “OECD Glossary of Statistical Terms - Research and development – UNESCO Definition,” 2001. https://stats.oecd.org/glossary/detail.asp?ID=2312, accessed 2021-04-26.

[8] ISO Central Secretary, “ISO 15288:2015(E): Systems and software engineering - System life cycle processes,” standard, International Organization for Standardization, Geneva, Switzerland, 2015.

[9] M. W. Maier, “Architecting principles for systems-of-systems,” *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998. https://doi.org/10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D.

[10] SEBoK Authors, “Systems engineering overview v. 2.2 released 15 may 2020,” 2021.

[11] K. Cormican, S. Morley, and P. Folan, “An analysis of virtual team characteristics: A model for virtual project managers,” *Journal of Technology Management & Innovation*, vol. 10, no. 1, pp. 188–203, 2015. http://dx.doi.org/10.4067/S0718-27242015000100014.

[12] E. Honoré-Livermore, "CubeSats in University: Using Systems Engineering Tools to Improve Reviews and Knowledge Management," *Procedia Computer Science*, vol. 153, pp. 63–70, 2019.

[13] E. Honoré-Livermore and C. Haskins, "Model-Based Systems Engineering for CubeSat FMECA," in *Recent Trends and Advances in Model Based Systems Engineering* (A. Madni, B. Boehm, D. Erwin, M. Moghaddam, M. Sievers, and M. Wheaton, eds.), Springer, Cham., 2022. `https://doi.org/10.1007/978-3-030-82083-1_45`.

[14] E. Honoré-Livermore, S. Bakken, and E. F. Prentice, "Factors Influencing the Development Time from TRL4 to TRL8 for CubeSat Subsystems at a University," in *9th International Systems & Concurrent Engineering for Space Applications Conference*, 2020.

[15] E. Honoré-Livermore and R. Birkeland, "Managing Product Development and Integration of a University CubeSat in a Locked down World," in *2021 IEEE Aerospace Conference (50100)*, pp. 1–12, 2021. `https://doi.org/10.1109/AERO50100.2021.9438490`.

[16] E. Honoré-Livermore, R. Birkeland, S. Bakken, J. L. Garrett, and C. Haskins, "Digital Engineering Management in an Academic CubeSat Project," *Special Issue on Systems Engineering Challenges in Journal of Aerospace Information Systems*, 2021. `https://doi.org/10.2514/1.I010981`.

[17] E. Honoré-Livermore, R. Lyells, J. L. Garrett, R. R. Angier, and B. Epps, "An Agile Systems Engineering Analysis of a University-built CubeSat," in *IS2021 INCOSE International Symposium on Systems Engineering*, vol. 31, pp. 1334–1348, 2021. `https://doi.org/10.1002/j.2334-5837.2021.00904.x`.

[18] E. Honoré-Livermore, R. Birkeland, and C. Haskins, "Addressing the Sustainable Development Goals with a System-of-Systems for Monitoring Arctic Coastal Regions," in *INCOSE International Symposium (ISSN 2334-5837)*, vol. 30, pp. 604–619, 2020. `http://dx.doi.org/10.1002/j.2334-5837.2020.00743.x`.

[19] E. Honoré-Livermore, A. Dallolio, R. Birkeland, D. D. Langer, C. Haskins, and T. A. Johansen, "Mbse modeling of a sos with a small satellite and autonomous surface vessels for persistent coastal monitoring," in *2021 16th International Conference of System of Systems Engineering (SoSE)*, pp. 156–161, 2021. `https://doi.org/10.1109/SOSE52739.2021.9497470`.

[20] E. Honoré-Livermore, K. R. Fossum, and E. Veitch, "Academics' Perception of Systems Engineering and Applied Research Projects," *Systems Engineering*, pp. 1–16, 2021. `https://doi.org/10.1002/sys.21599`.

[21] K. R. Fossum, E. Honoré-Livermore, E. Veitch, and C. Haskins, "Systems Engineering and Project Management in Academia: The Role of Complexity Dimensions," *Submitted to Systems Engineering (The Journal of The International Council on Systems Engineering), Wiley Periodicals*, 2021.

[22] M. Cho and H. Mazui, "Best practices for successful lean satellite projects," in *4th UNISEC Global Meeting*, (Varna, Bulgaria), 2016.

[23] M. Langer and J. Bouwmeester, "Reliability of cubesats – statistical data, developers' beliefs and the way forward," in *Proceedings of the AIAA/USU Conference on Small Satel-*

lites, 2016. SSC16-TS10ADVTECH2-4, https://digitalcommons.usu.edu/smallsat/2016/TS10AdvTech2/4/.

[24] P. Faure, A. Tanaka, and M. Cho, "Toward lean satellites reliability improvement using horyu-iv project as case study," *Acta Astronautica*, vol. 133, pp. 33–49, 2017. https://doi.org/10.1016/j.actaastro.2016.12.030.

[25] M. Swartout, "Cubesat mission success: Are we getting better?," in *2019 CubeSat Developers' Workshop*, (San Luis Opisbo, CA), 2019.

[26] M. Langer, C. Olthoff, J. Harder, C. Fuchs, M. Dziura, A. Hoehn, and U. Walter, "Results and lessons learned from the cubesat mission first-move," in *Small Satellite Missions for Earth Observation, 10th International Symposium, IAA*, (Berlin), 2015.

[27] J. Grande, R. Birkeland, A. Gjersvik, S. Mathisen, and C. Stausland, "Norwegian student satellite program – lessons learned," in *68th International Astronautical Congress*, 2017.

[28] J. Larsen and J. D. Nielsen, "Development of cubesats in an educational context," in *International Conference on Recent Advances in Space Technologies*, (Istanbal, Turkey), pp. 777–782, IEEE, 2011. https://doi.org/10.1109/RAST.2011.5966948.

[29] K. Rajan, T. A. Johansen, A. Sørensen, and R. Birkeland, "A Roadmap for a SmallSat Program," 2017. Internal strategic document.

[30] M. W. Maier, "Architecting a portfolio of systems," *Systems Engineering*, vol. 22, no. 4, pp. 335–347, 2019. https://doi.org/10.1002/sys.21486.

[31] IOCCG, *Earth Observations in Support of Global Water Quality Monitoring.*, vol. 17 of *Reports of the International Ocean-Colour Coordinating Group*. Darthmouth, Canada: IOCCG, 2018.

[32] IOCCG, *Why Ocean Colour? The Societal Benefits of Ocean-Colour Technology*, vol. 7 of *Reports of the International Ocean Colour Coordinating Group*. Dartmouth, Canada: IOCCG, 2008.

[33] G. Quintana-Diaz, R. Birkeland, T. Ekman, and E. Honoré-Livermore, "An sdr mission measuring uhf signal propagation and interference between small satellites in leo and arctic sensors," in *Proceedings of the Small Satellite Conference*, (Logan, UT), 2019. SSC19-WP2-01, https://digitalcommons.usu.edu/smallsat/2019/all2019/1/.

[34] M. E. Grøtte, R. Birkeland, E. Honoré-Livermore, S. Bakken, J. L. Garrett, E. F. Prentice, F. Sigernes, M. Orlandić, J. T. Gravdahl, and T. A. Johansen, "Ocean Color Hyperspectral Remote Sensing with High Resolution and Low Latency – the HYPSO-1 CubeSat Mission," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–19, 2021. https://doi.org/10.1109/TGRS.2021.3080175.

[35] A. Dallolio, G. Quintana-Diaz, E. Honoré-Livermore, J. L. Garrett, R. Birkeland, and T. A. Johansen, "A satellite-usv system for persistent observation of mesoscale oceanographic phenomena," *Remote Sensing*, vol. 13, no. 16, p. 3229, 2021.

[36] A. G. C. Guerra, F. Francisco, J. Villate, F. Aguado Agelet, O. Bertolami, and K. Rajan, "On small satellites for oceanography: A survey," *Acta Astronautica*, vol. 127, pp. 404–423, 2016. https://doi.org/10.1016/j.actaastro.2016.06.007.

[37] Z. Volent, G. Johnsen, E. Hovland, A. Folkestad, L. Olsen, K. Tangen, and K. Sorensen, "Improved monitoring of phytoplankton bloom dynamics in a norwegian fjord by integrating satellite data, pigment analysis, and ferrybox data with a coastal observation network," *Journal of Applied Remote Sensing*, vol. 5, no. 1, p. 053561, 2011.

[38] J. B. d. Sousa, J. Pereira, J. Pinto, L. Paulo Claro, G. João Madaleno, J. Fontes, M. Silva, K. Rajan, T. A. Johansson, J. Alves, A. Munafò, K. Pelekanakis, R. Petroccia, M. C. Silva, and M. Incze, "Rapid environmental picture atlantic exercise 2015: A field report," in *OCEANS 2016 MTS/IEEE Monterey*, pp. 1–6, 2016. https://doi.org/10.1109/OCEANS.2016.7761496.

[39] A. S. Ferreira, M. Costa, F. Py, J. Pinto, M. A. Silva, A. Nimmo-Smith, T. A. Johansen, J. B. de Sousa, and K. Rajan, "Advancing multi-vehicle deployments in oceanographic field experiments," *Autonomous Robots*, vol. 43, no. 6, pp. 1555–1574, 2019. https://doi.org/10.1007/s10514-018-9810-x.

[40] T. B. Curtin and J. G. Bellingham, "Progress toward autonomous ocean sampling networks," *Deep Sea Research Part II: Topical Studies in Oceanography*, vol. 56, no. 3, pp. 62–67, 2009. https://doi.org/10.1016/j.dsr2.2008.09.005.

[41] J. Das, F. Py, T. Maughan, T. O'Reilly, M. Messié, J. Ryan, G. S. Sukhatme, and K. Rajan, "Coordinated sampling of dynamic oceanographic features with underwater vehicles and drifters," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 626–646, 2012. https://doi.org/10.1177/0278364912440736.

[42] C. Araguz, E. Bou-Balust, and E. Alarcón, "Applying autonomy to distributed satellite systems: Trends, challenges, and future prospects," *Systems Engineering*, vol. 21, no. 5, pp. 401–416, 2018. https://doi.org/10.1002/sys.21428.

[43] E. Ordoukhanian and A. Madni, "Model-based approach to engineering resilience in multi-uav systems," *Systems*, vol. 7, no. 1, 2019. https://doi.org/10.3390/systems7010011.

[44] M. LaSorda, J. M. Borky, and R. M. Sega, "Model-based architecture and programmatic optimization for satellite system-of-systems architectures," *System Engineering*, vol. 21, 2018. https://doi.org/10.1002/sys.21444.

[45] C. Guariniello, L. Mockus, A. K. Raz, and D. A. DeLaurentis, "Towards intelligent architecting of aerospace system-of-systems," in *2019 IEEE Aerospace Conference*, pp. 1–11, 2019. https://doi.org/10.1109/AERO.2019.8742173.

[46] E. F. Prentice, M. E. Grøtte, F. Sigernes, and T. A. Johansen, "Design of a hyperspectral imager using cots optics for small satellite applications," in *International Conference on Space Optics*, (Virtual), 2021.

[47] M. Hov, *Design and Implementation of Hardware and Software Interfaces for a Hyperspectral Payload in a Small Satellite*. PhD thesis, Norwegian University of Science and Technology, 2019.

[48] P. J. Bentley, M. Gulbrandsen, and S. Kyvik, "The relationship between basic and applied research in universities," *Higher Education*, vol. 70, pp. 689–709, 2015. http://dx.doi.org/10.1007/s10734-015-9861-2.

[49] N. Fowler, M. Lindahl, and D. Skjöld, "The projectification of university research," *International Journal of Managing Projects in Business*, vol. 8, no. 1, pp. 9–32, 2015. https://doi.org/10.1108/IJMPB-10-2013-0059.

[50] R. Birkeland and O. Gutteberg, "Educational aspects of a cubesat project," in *2nd IAA Conference on University Satellite Missions and CubeSat Workshop*, 2013.

[51] L. Berthoud, M. Swartout, J. Cutler, D. Klumpar, J. A. Larsen, and J. D. Nielsen, "University cubesat project management for success," in *Proceedings of the AIAA/USU Conference on Small Satellites*, (Logan, UT, United States), 2019. SSC19-S3-63, https://digitalcommons.usu.edu/smallsat/2019/all2019/63/.

[52] P. M. B. Waswa and S. Redkar, "A survey of space mission architecture and system actualisation methodologies," *International Journal of Aerospace Engineering*, vol. 4, no. 3, pp. 234–252, 2017. https://doi.org/10.1504/IJSPACESE.2017.085674.

[53] California Polytechnic's PolySat Program, *CubeSat 101*. NASA's CubeSat Launch Initiative, California Polytechnic State University at San Luis Obispo, 2017.

[54] ISO Central Secretary, "ISO19683 Space systems - Design qualification and acceptance tests of small spacecraft and units," standard, International Organization for Standardization, Geneva, Switzerland, 2017.

[55] ECSS Secretariat, "Tailored ecss engineering standards for in-orbit demonstration cubesat projects," tech. rep., European Space Agency, 2016-2017.

[56] I. Uwarowa, P. Brunne, A. Budzynski, M. Gawin, G. Gajoch, M. Gumiela, P. Kuligowski, A. Lukasik, D. Roszkowski, E. Ryszawa, and M. Sobiecki, "Pw-sat2 satellite lessons learned," in *Proceedings of the International Astronautical Congress, IAC*, vol. 2019-October, 2019.

[57] G. Loureiro, W. F. Panades, and A. Silva, "Lessons learned in 20 years of application of systems concurrent engineering to space products," *Acta Astronautica*, vol. 151, pp. 44–52, 2018.

[58] B. Segret, A. Semery, J. Vannitsen, B. Mosser, J. J. Miau, J. C. Juang, and F. Deleflie, "The paving stones: Initial feed-back on an attempt to apply the agile principles for the development of a cubesat space mission to mars," in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 9150, 2014.

[59] L. Alminde, M. Bisgaard, D. D. V. Bhanderi, and J. D. Nielsen, "Experience and methodology gained from 4 years of student satellite projects," in *2nd International Conference on Recent Advances in Space Technologies*, vol. 1, (Turkey, Istanbul), pp. 94–99, IEEE Xplore, 2005. https://doi.org/10.1109/RAST.2005.1512542.

[60] J. Bouwmeester, G. Aalbers, and W. Ubbels, "Preliminary Mission Results and Project Evaluation of the Delfi-C3 Nano-Satellite," in *4S Symposium Small Satellites Systems and Services*, 2008.

[61] C. D. Abadi and T. Bahill, "Product and process: an innovative approach for a student satellite program from the systems engineering point of view," in *15th Annual/USU Conference on Small Satellites*, 2001. http://tucson.sie.arizona.edu/sysengr/SSP/uasat/document/AIAAPaper.pdf.

[62] INCOSE, *INSIGHT: Model-Based Systems Engineering: A New Paradigm*, vol. 1. THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING, 1998.

[63] INCOSE, "Introduction to this special edition on model-based systems engineering," *INSIGHT: Model-Based Systems Engineering: The New Paradigm*, vol. 12, no. 4, 2009.

[64] E. W. Brower, C. Delp, R. Karban, M. Piette, I. Gomes, and E. v. Wyk, "OpenCAE Case Study: Europa Lander Concept," in *INCOSE International Workshop*, (Torrance, CA.), 2019.

[65] S. Bonnet, J.-L. Voirin, and J. Navas, "Augmenting requirements with models to improve the articulation between system engineering levels and optimize v&v practices," *INCOSE International Symposium*, vol. 29, no. 1, pp. 1018–1033, 2019. `https://doi.org/10.1002/j.2334-5837.2019.00650.x`.

[66] J. Navas, S. Bonnet, J.-L. Voirin, and G. Journaux, "Models as enablers of agility in complex systems engineering," in *INCOSE International Symposium*, vol. 30, pp. 339–355, Year of Conference.

[67] D. Kaslow and A. Madni, "Validation and Verification of MBSE-compliant CubeSat Reference Model," in *Conference on Systems Engineering Research* (B. B. Azad M. Madni, ed.), (Redondo Beach, CA, USA), 2017. `https://doi.org/10.1007/978-3-319-62217-0_27`.

[68] Y. Menshenin, K. Latyshev, A. Mahfouz, and D. Pritykin, "Applying the model-based conceptual design approach to adcs development: A case study of a cubesat swarm mission," in *Proceedings of the International Astronautical Congress, IAC*, vol. 2019-October, 2019.

[69] D. Kaslow, B. Ayres, P. T. Cahill, and L. Hart, "A model-based systems engineering approach for technical measurement with application to a cubesat," in *IEEE Aerospace Conference Proceedings*, vol. 2018-March, pp. 1–10, 2018. `https://doi.org/10.1109/AERO.2018.8396443`.

[70] D. Kaslow, P. T. Cahill, and B. Ayres, "Development and application of the cubesat system reference model," in *IEEE Aerospace Conference Proceedings*, 2020. `https://doi.org/10.1109/AERO47225.2020.9172714`.

[71] R. Gomes, J. Straub, A. Jones, J. Morgan, S. Tipparach, A. Sletten, K. W. Kim, D. Loegering, N. Feikema, K. Dayananda, G. Miryala, A. Gass, K. Setterstrom, J. Mischel, D. Shipman, and C. Nazzaro, "An interconnected network of uas as a system-of-systems," in *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, pp. 1–7, 2017.

[72] F. Pérez-Lissi, F. Aguado-Agelet, A. Vázquez, P. Yañez, P. Izquierdo, S. Lacroix, R. Bailon-Ruiz, J. Tasso, A. Guerra, and M. Costa, "FIRE-RS: Integrating land sensors, cubesat communications, unmanned aerial vehicles and a situation assessment software for wildland fire characterization and mapping," in *69th International Astronautical Congress*, 2018.

[73] A. M. Madni and M. Sievers, "Systems integration: Key perspectives, experiences, and challenges," *Systems Engineering*, vol. 17, no. 1, pp. 37–51, 2014. `https://doi.org/10.1002/sys.21249`.

[74] International Council of Systems Engineering, "Incose systems of systems primer incose-tp-2018-003-01.0," tech. rep., INCOSE, 2018.

[75] A. M. Madni and M. Sievers, “System of systems integration: Fundamental concepts, challenges and opportunities,” in *Advances in Systems Engineering* (J. Hsu and R. Curran, eds.), vol. 252, pp. 1–34, Reston, Virginia, USA: American Institute of Aeronautics and Astronautics, 2016.

[76] J. T. J. Mathieson, T. Mazzuchi, and S. Sarkani, “The systems engineering devops lemniscate and model-based system operations,” *IEEE Systems Journal*, pp. 1–12, 2020. `https://doi.org/10.1109/JSYST.2020.3015595`.

[77] S. Paluch, D. Antons, M. Brettel, C. Hopp, T.-O. Salge, F. Piller, and D. Wentzel, “Stage-gate and agile development in the digital age: Promises, perils, and boundary conditions,” *Journal of Business Research*, vol. 110, pp. 495–501, 2020. `https://doi.org/10.1016/j.jbusres.2019.01.063`.

[78] J. Campos and P. Ferguson, “Decentralized project management concept for schedule-critical space projects,” *Aerospace Systems*, 2021.

[79] R. Dove and B. Schindel, “Agile systems engineering life cycle model for mixed discipline engineering,” in *INCOSE International Symposium 2019*, vol. 29, (Orlando, Florida), pp. 86–104, 2019. `https://doi.org/10.1002/j.2334-5837.2019.00591.x`.

[80] ECSS Secretariat, “Space engineering: Agile software development handbook,” tech. rep., European Space Agency, 2020.

[81] A. Atzberger and K. Paetzold, “Current challenges of agile hardware development: What are still the pain points nowadays?,” *Proceedings of the Design Society: International Conference on Engineering Design*, vol. 1, no. 1, pp. 2209–2218, 2019.

[82] V. Berg, J. Birkeland, A. Nguyen-Duc, I. O. Pappas, and L. Jaccheri, “Achieving agility and quality in product development - an empirical study of hardware startups,” *Journal of Systems and Software*, vol. 167, p. 110599, 2020.

[83] N. Garzaniti, S. Briatore, C. Fortin, and A. Golkar, “Effectiveness of the scrum methodology for agile development of space hardware,” in *2019 IEEE Aerospace Conference*, pp. 1–8, 2019. `https://doi.org/10.1109/AERO.2019.8741892`.

[84] J. N. Martin, “3.1.2 the seven samurai of systems engineering: Dealing with the complexity of 7 interrelated systems,” *INCOSE International Symposium*, vol. 14, no. 1, pp. 459–470, 2004.

[85] G. Quintana-Diaz, T. Ekman, J. M. Lago Agra, D. Hurtado de Mendoza, A. González Muíño, and F. Aguado Agelet, “In-orbit measurements and analysis of radio interference in the uhf amateur radio band from the lume-1 satellite,” *Remote Sensing*, vol. 13, no. 16, p. 3252, 2021.

[86] L. Rising and N. S. Janoff, “The scrum software development process for small teams,” *IEEE Software*, vol. 17, no. 4, pp. 26–32, 2000. `https://doi.org/10.1109/52.854065`.

[87] M. A. Bone, M. R. Blackburn, D. H. Rhodes, D. N. Cohen, and J. A. Guerrero, “Transforming systems engineering through digital engineering,” *The Journal of Defense Modeling and Simulation*, vol. 16, no. 4, pp. 339–355, 2019. `https://doi.org/10.1177/1548512917751873`.

[88] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, and R. Twiggs, "Cubesat: A new generation of picosatellite for education and industry low-cost space experimentation," in *Proceedings of the AIAA/USU Conference on Small Satellites*, (UT, USA), 2000. SSC00-TS5-32, https://digitalcommons.usu.edu/smallsat/2000/All2000/32/.

[89] Kepler, "Kepler's first ku-band satellite is in orbit." Accessed February 2018, http://www.keplercommunications.com/blog/item/first-launch.

[90] Planet, "Planet webpage," 2016. Accessed April 2016, www.planet.com.

[91] B. Boehm, R. Valerdi, and E. Honour, "The roi of systems engineering: Some quantitative results for software-intensive systems," *Systems Engineering*, vol. 11, no. 3, pp. 221–234, 2008. https://doi.org/10.1002/sys.20096.

[92] W. J. Larson and J. R. Wertz, *Space mission analysis and design*. Space technology library, Torrance, Calif: Microcosm Press, 3rd ed. ed., 1999.

[93] Y. M. D. Pereira, A. A. Brandão, A. S. Costa, B. C. Junqueira, S. G. Guimarães, and G. Loureiro, "Lessons learned on systems of systems engineering: Systems concurrent engineering of a constellation of cubesat formations," in *Proceedings of the International Astronautical Congress, IAC*, 2019.

[94] H. Reza, C. Korvald, J. Straub, J. Hubber, N. Alexander, and A. Chawla, "Toward requirements engineering of cyber-physical systems: Modeling cubesat," in *2016 IEEE Aerospace Conference*, pp. 1–13, 2016. https://doi.org/10.1109/AERO.2016.7500897.

[95] A. Lill, D. Messmann, and M. Langer, "Agile software development for space applications," in *Deutscher Luft- und Raumfahrtkongress 2017*, 2017. https://www.dglr.de/publikationen/2017/450309.pdf.

[96] R. Santos, F. Flentge, M.-E. Begin, and V. Navarro, "Agile technical management of industrial contracts: Scrum development of ground segment software at the european space agency," in *Agile Processes in Software Engineering and Extreme Programming*, (Berlin, Heidelberg), pp. 290–305, Springer Berlin Heidelberg, 2011.

[97] S. B. Johnson, *The Secret of Apollo: Systems Management in American and European Space Programs*. Johns Hopkins University Press, 2006.

[98] J. H. Brill, "Systems engineering— a retrospective view," *Systems Engineering*, vol. 1, no. 4, pp. 258–266, 1998. https://doi.org/10.1002/(SICI)1520-6858(1998)1:4<258::AID-SYS2>3.0.CO;2-E.

[99] A. Kossiakoff, W. N. Sweet, S. J. Seymour, and S. M. Biemer, *Systems Engineering Principles and Practice*, vol. 83. John Wiley & Sons, 2011.

[100] SEBoK, "Systems engineering overview — sebok,," 2020. [Online; accessed 17-February-2021].

[101] M. Winter and P. Checkland, "Soft systems: a fresh perspective for project management," *Proceedings of the Institution of Civil Engineers - Civil Engineering*, vol. 156, no. 4, pp. 187–192, 2003. https://doi.org/10.1680/cien.2003.156.4.187.

[102] L. Crawford and J. Pollack, "Hard and soft projects: a framework for analysis," *International Journal of Project Management*, vol. 22, no. 8, pp. 645–653, 2004. https://doi.org/

10.1016/j.ijproman.2004.04.004.

[103] SEBoK Authors, "System approaches," 2000.

[104] S. C. Cook and S. A. Wilson, "The enduring path to system success: Investment in quality early-phase systems engineering," in *INCOSE International Symposium*, vol. 29, (Orlando, FL, USA), pp. 817–835, 2019. https://doi.org/10.1002/j.2334-5837.2019.00637.x.

[105] K. T. Yeo, "Systems thinking and project management — time to reunite," *International Journal of Project Management*, vol. 11, no. 2, pp. 111–117, 1993. https://doi.org/10.1016/0263-7863(93)90019-J.

[106] M. Winter, "Problem structuring in project management: an application of soft systems methodology (SSM)," *Journal of the Operational Research Society*, vol. 57, no. 7, pp. 802–812, 2006. https://doi.org/10.1057/palgrave.jors.2602050.

[107] F. Vanek, P. Jackson, and R. Grzybowski, "Systems engineering metrics and applications in product development: A critical literature review and agenda for further research," *Systems Engineering*, vol. 11, no. 2, pp. 107–124, 2008. https://doi.org/10.1002/sys.20089.

[108] J. P. Elm and D. R. Goldenson, "The business case for systems engineering study: Results of the systems engineering effectiveness survey," Tech. Rep. ADA610517, CARNEGIE-MELLON UNIV. PITTSBURGH PA SOFTWARE ENGINEERING INST, 2012.

[109] E. Honour, "Understanding the value of systems engineering," in *INCOSE International Symposium*, vol. 14, (Toulouse, France), pp. 1208–1222, 2004. https://doi.org/10.1002/j.2334-5837.2004.tb00567.x.

[110] E. Honour, "Systems engineering return on investment," in *INCOSE International Symposium*, vol. 20, (Chicago, IL, USA), pp. 1422–1439, 2010. https://doi.org/10.1002/j.2334-5837.2010.tb01150.x.

[111] S. A. Sheard, H. Lykins, and J. R. Armstrong, "Overcoming barriers to systems engineering process improvement," *Systems Engineering*, vol. 3, no. 2, pp. 59–67, 2000. https://doi.org/10.1002/1520-6858(2000)3:2<59::AID-SYS1>3.0.CO;2-B.

[112] B. Heydari and M. J. Pennock, "Guiding the behavior of sociotechnical systems: The role of agent-based modeling," *Systems Engineering*, vol. 21, no. 3, pp. 210–226, 2018. https://doi.org/10.1002/sys.21435.

[113] L. Pajerek, "Processes and organizations as systems: When the processors are people, not pentiums," *Systems Engineering*, vol. 3, no. 2, pp. 103–111, 2000. https://doi.org/10.1002/1520-6858(2000)3:2<103::AID-SYS5>3E3.0.CO;2-D.

[114] W. Donaldson, "In praise of the "ologies": A discussion of and framework for using soft skills to sense and influence emergent behaviors in sociotechnical systems," *Systems Engineering*, vol. 20, no. 5, pp. 467–478, 2017. https://doi.org/10.1002/sys.21408.

[115] C. J. White and B. L. Mesmer, "Research needs in systems engineering: Report from a University of Alabama in Huntsville workshop," *Systems Engineering*, 2019. https://doi.org/10.1002/sys.21501.

[116] W. B. Rouse, M. J. Pennock, and J. Cardoso, "Advances in sociotechnical systems," in *Advances in Systems Engineering* (J. Hsu and R. Curran, eds.), vol. 252, p. 30, Reston, Virginia, USA: American Institute of Aeronautics and Astronautics, 2016.

[117] D. Vaughan, *The Challenger launch decision : risky technology, culture, and deviance at NASA*. Chicago: University of Chicago Press, 1996.

[118] Z. Szajnfarber and E. Gralla, "Qualitative methods for engineering systems: Why we need them and how to use them," *Systems Engineering*, vol. 20, no. 6, pp. 497–511, 2017. `https://doi.org/10.1002/sys.21412`.

[119] T. McDermott, M. Nadolski, D. Duncan, and M. Clowse, "Modeling the sociotechnical dimensions of urban resilience: Community-level microgrids," *INCOSE International Symposium*, vol. 28, no. 1, pp. 743–759, 2018. `https://doi.org/10.1002/j.2334-5837.2018.00513.x`.

[120] M. W. Maier, "The role of modeling and simulation in system of systems development," in *Modeling and Simulation Support for System of Systems Engineering Applications* (L. Rainey and A. Tolk, eds.), pp. 11–41, John Wiley & Sons, Inc., 2015.

[121] J. Dahmann, G. Rebovich, J. Lane, R. Lowry, and K. Baldwin, "An implementers' view of systems engineering for systems of systems," in *2011 IEEE International Systems Conference*, pp. 212–217, 2011. `https://doi.org/10.1109/SYSCON.2011.5929039`.

[122] R. M. Verburg, P. Bosch-Sijtsema, and M. Vartiainen, "Getting it done: Critical success factors for project managers in virtual work settings," *International Journal of Project Management*, vol. 31, no. 1, pp. 68–79, 2013. `https://doi.org/10.1016/j.ijproman.2012.04.005`.

[123] J. Olaisen and O. Revang, "Working smarter and greener: Collaborative knowledge sharing in virtual global project teams," *International Journal of Information Management*, vol. 37, no. 1, Part A, pp. 1441–1448, 2017. `https://doi.org/10.1016/j.ijinfomgt.2016.10.002`.

[124] N. Drouin, M. Bourgault, and C. Gervais, "Effects of organizational support on components of virtual project teams," *International Journal of Managing Projects in Business*, vol. 3, pp. 625–641, 2010. `https://doi.org/10.1108/17538371011076082`.

[125] T. Kayworth and D. Leidner, "The global virtual manager: a prescription for success," *European Management Journal*, vol. 18, no. 2, pp. 183–194, 2000. `https://doi.org/10.1016/S0263-2373(99)00090-0`.

[126] B. Sauser, J. Boardman, and A. Gorod, "System of systems management," in *System of Systems Engineering: Innovations for the 21st Century* (D. M. Jamshidi, ed.), John Wiley and Sons, Inc., 2008.

[127] P. J. Componation, A. D. Youngblood, D. R. Utley, and P. A. Farrington, "A preliminary assessment of the relationships between project success, system engineering, and team organization," *Engineering Management Journal*, vol. 20, no. 4, pp. 40–46, 2008. `https://doi.org/10.1080/10429247.2008.11431787`.

[128] M. Malbert, D. Estève, C. Baron, P. Esteban, and R. Xue, "Managing requirements: For an integrated approach from system engineering to project management," in *Advances in*

Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World, (Berlin, Heidelberg), pp. 142–149, Springer Berlin Heidelberg, 2014.

[129] S. Kordova, E. Katz, and M. Frank, "Managing development projects—the partnership between project managers and systems engineers," *Systems Engineering*, vol. 22, no. 3, pp. 227–242, 2019. https://doi.org/10.1002/sys.21474.

[130] M. Emes, A. Smith, and D. Cowper, "Confronting an identity crisis—how to "brand" systems engineering," *Systems Engineering*, vol. 8, no. 2, pp. 164–186, 2005. https://doi.org/10.1002/sys.20028.

[131] E. Conforto, M. Rossi, E. Rebentisch, J. Oehmen, and M. Pacenza, "Survey report: Improving integration of program management and systems engineering," in *23rd INCOSE Annual International Symposium*, (Philadelphia, US), 2013.

[132] J. Oehmen, B. W. Oppenheim, D. Secor, E. Norman, E. Rebentisch, J. A. Sopko, M. Steuber, R. Dove, K. Moghaddam, S. McNeal, M. Bowie, M. Ben-Daya, W. Altman, and J. Driessnack, "The Guide to Lean Enablers for Managing Engineering Programs," 2012. https://dspace.mit.edu/handle/1721.1/70495.

[133] T. Srivastava, J.-C. Roussel, and J. Lomax, "Minutes - april incose pm-se integration working group meeting," 2021. Internal presentation at the INCOSE Project Management - Systems Engineering Working Group Meeting April 2021.

[134] B. Lilburn, "Integrating systems engineering and program management," in *INCOSE International Symposium*, vol. 6, pp. 689–696, 1996. https://doi.org/10.1002/j.2334-5837.1996.tb02071.x.

[135] T. R. Browning, "Applying the design structure matrix to system decomposition and integration problems: a review and new directions," *IEEE Transactions on Engineering Management*, vol. 48, no. 3, pp. 292–306, 2001. https://doi.org/10.1109/17.946528.

[136] C. L. Roe, "Project Management and Systems Engineering in an IPD Environment," in *INCOSE International Symposium*, vol. 6, (Boston, MA), pp. 1063–1068, 1996. https://doi.org/10.1002/j.2334-5837.1996.tb02123.x.

[137] B. Smith and J. van Gaasbeek, "Project Management, Systems Engineering and the Amazing DNA Double Helix," in *INCOSE International Symposium*, vol. 6, pp. 882–886, 1996. https://doi.org/10.1002/j.2334-5837.1996.tb02098.x.

[138] S. B. Johnson, "Three approaches to big technology: Operations research, systems engineering, and project management," *Technology and Culture*, vol. 38, no. 4, pp. 891–919, 1997. https://www.jstor.org/stable/3106953.

[139] H. Mooz and K. Forsberg, "Visualizing system engineering and project management as an integrated process," in *INCOSE International Symposium*, vol. 7, (Los Angeles, CA), pp. 573–580, 1997. https://doi.org/10.1002/j.2334-5837.1997.tb02221.x.

[140] A. Sharon, V. Perelman, D. Dori, and P. A, "A project-product lifecycle management approach for improved systems engineering practices," in *INCOSE International Symposium*, vol. 18, pp. 942–957, 2008. https://doi.org/10.1002/j.2334-5837.2008.tb00854.x.

[141] B. G. Barker and D. Verma, "Systems engineering effectiveness: A complexity point paradigm for software intensive systems in the information technology sector," *Engineering Management Journal*, vol. 15, no. 3, pp. 29–35, 2003. https://doi.org/10.1080/10429247.2003.11415213.

[142] R. Xue, C. Baron, P. Esteban, D. Esteve, and M. Malbert, "Towards the success of design projects by the alignment of processes in collaborative engineering," in *A Joint conference on mechanical design engineering and advanced manufacturing*, 2014.

[143] L. Zheng, C. Baron, P. Esteban, R. Xue, Q. Zhang, and S. Yang, "Using leading indicators to improve project performance measurement," *Journal of Systems Science and Systems Engineering*, vol. 28, no. 5, pp. 529–554, 2019. https://doi.org/10.1007/s11518-019-5414-z.

[144] R. Xue, C. Baron, P. Esteban, J. B. Yang, and L. Zheng, "Toward an improved monitoring of engineering projects," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 10, pp. 3541–3553, 2020. https://doi.org/10.1109/TSMC.2018.2884196.

[145] R. Xue, C. Baron, R. Vingerhoeds, and P. Esteban, "Enhancing engineering project management through process alignment," *Engineering Management Journal*, pp. 1–19, 2021. https://doi.org/10.1080/10429247.2020.1865002.

[146] A. Jaafari and K. Manivong, "Towards a smart project management information system," *International Journal of Project Management*, vol. 16, no. 4, pp. 249–265, 1998. https://doi.org/10.1016/S0263-7863(97)00037-9.

[147] M. Malik, S. Sarwar, and S. Orr, "Agile practices and performance: Examining the role of psychological empowerment," *International Journal of Project Management*, 2020. https://doi.org/10.1016/j.ijproman.2020.09.002.

[148] M. Bianchi, G. Marzi, and M. Guerini, "Agile, stage-gate and their combination: Exploring how they relate to performance in software development," *Journal of Business Research*, vol. 110, pp. 538–553, 2020. https://doi.org/10.1016/j.jbusres.2018.05.003.

[149] M. Rother, Mike Rosenthal, "An approach to becoming agile in a dynamic world. helping employees develop scientific thinking empowers them to solve problems and make decisions," *Association for Manufacturing Excellence: TARGET*, vol. 34, 2018.

[150] L. Davidovitch, A. Parush, and A. Shtub, "Simulator-based team training to share resources in a matrix structure organization," *IEEE Transactions on Engineering Management*, vol. 57, no. 2, pp. 288–300, 2010. https://doi.org/10.1109/TEM.2009.2023142.

[151] I. Cohen, M. Iluz, and A. Shtub, "A simulation-based approach in support of project management training for systems engineers," *Systems Engineering*, vol. 17, no. 1, pp. 26–36, 2014. https://doi.org/10.1002/sys.21248.

[152] M. Törngren and U. Sellgren, "Complexity challenges in development of cyber-physical systems," in *Principles of Modeling. Lecture Notes in Computer Science* (M. S. Martin Lohstroh, Patricia Derler, ed.), pp. 478–503, Switzerland: Springer, 2018. https://doi.org/10.1007/978-3-319-95246-8_27.

[153] T. McDermott, E. Van Aken, N. Hutchison, M. Blackburn, M. Clifford, N. Chean, A. Salado, and K. Henderson, "Summary report task order wrt-1001: Digital engineering metrics," tech. rep., Stevens Institute of Technology, 2020.

[154] W. Len, J. Brown, A. Toorian, R. Coelho, L. Brooks, J. P. Suari, and R. Twiggs, "Cubesat development in education and into industry," in *AIAA Space*, vol. 2, (San Jose, California), 2006. `https://doi.org/10.2514/6.2006-7296`.

[155] Y. Tsuda, N. Sako, T. Eishima, T. Ito, Y. Arikawa, N. Miyamura, R. A. A. Tanaka, and S. Nakasuka, "University of tokyo's cubesat project: Its educational and technological significance," in *15th AnnualAIAA/USU Conference on Small Satellites*, 2001.

[156] Z. S. Decker, *A Systems-Engineering Assessment of Multiple CubeSat Build Approaches*. PhD thesis, Massachusetts Institute of Technology (MIT), 2016.

[157] C. C. Venturini, B. M. Braun, D. Hinkley, and G. K. Berg, "Improving mission success of cubesats," tech. rep., Space and Missile Systems Center, 2018.

[158] I. Latachi, T. Rachidi, M. Karim, and A. Hanafi, "Reusable and reliable flight-control software for a fail-safe and cost-efficient cubesat mission: Design and implementation," *Aerospace*, vol. 7, no. 10, pp. 1–29, 2020. `https://doi.org/10.3390/aerospace7100146`.

[159] E. Kulu, "Nanosats database," 2021. Accessed February 2021, `www.nanosats.eu`.

[160] M. Langer, M. Weisgerber, J. Bouwmeester, and A. Hoehn, "A reliability estimation tool for reducing infant mortality in cubesat missions," in *2017 IEEE Aerospace Conference*, pp. 1–9, 2017.

[161] A. Menchinelli, F. Ingiosi, L. Pamphili, P. Marzioli, R. Patriarca, F. Costantino, and F. Piergentili, "A reliability engineering approach for managing risks in cubesats," *Aerospace*, vol. 5, p. 121, 2018.

[162] A. Scholz, "Survey of international space standards with potential application for cubesat projects," in *International Workshop on Lean Satellite Standardization*, (Tokyo, Japan), 2017.

[163] T. Lock, "Standards for cubesats – a european space agency approach," in *International Workshop on Lean Satellite Standardization*, (Tokyo, Japan), 2017.

[164] M. Jacobs and D. Selva, "A cubesat catalog design tool for a multi-agent architecture development framework," in *IEEEAC*, 2015.

[165] J. Triana, S. Bautista, and F. Alexander Díaz González, "Identification of design considerations for small satellite remote sensing systems in low earth orbit," *Journal of Aerospace Technology and Management*, vol. 7, pp. 121–134, 2015.

[166] Y.-K. Chang, K.-L. Hwang, and S.-J. Kang, "Sedt (system engineering design tool) development and its application to small satellite conceptual design," *Acta Astronautica*, vol. 61, no. 7, pp. 676–690, 2007.

[167] A. McInnes, D. M Harps, J. A Lang, and C. M Swenson, "A systems engineering tool for small satellite design," in *Proceedings of the 15th Annual AIAA/USU Conference on Small Satellites*, 2001. SSC01-TS6-37, `https://digitalcommons.usu.edu/`

smallsat/2001/All2001/37/.

[168] M. Ehresmann, M. Fugmann, J. Skalden, D. Galla, C. Montag, G. Herdrich, S. Fasoulas, and S. Klinkner, “Iras: Progress in development of the digital concurrent engineering platform, software tools and innovative technologies,” in *Proceedings of the International Astronautical Congress, IAC*, 2019.

[169] B. M. Arnaut, G. Loureiro, D. D. Viot, E. E. Burger, L. L. Costa, and M. Franchitto, “Systems engineering process of a cubesat from the perspective of operations,” in *23rd Annual International Symposium of the International Council on Systems Engineering, INCOSE 2013*, vol. 1, pp. 105–117, 2013.

[170] B. W. Boehm, “Software engineering economics,” *IEEE Transactions of Software Engineering*, vol. 1, pp. 4–21, 1984.

[171] R. Dove and R. LaBarge, “Fundamentals of agile systems engineering – part 2,” in *INCOSE International Symposium*, vol. 24, pp. 876–892, 2014. https://doi.org/10.1002/j.2334-5837.2014.tb03187.x.

[172] D. K. Rigby, J. Sutherland, and H. Takeuchi, “Embracing agile: how to master the process that’s transforming management,” *Harvard Business Review*, vol. 94, no. 5, pp. 40–50, 2016.

[173] M. Bott and B. Mesmer, “An analysis of theories supporting agile scrum and the use of scrum in systems engineering,” *Engineering Management Journal*, vol. 32, no. 2, pp. 76–85, 2020. https://doi.org/10.1080/10429247.2019.1659701.

[174] R. Turner, “Toward agile systems engineering processes,” *The Journal of Defence Software Engineering*, vol. 20, pp. 11–15, 2007.

[175] M. A. G. Darrin and W. S. Devereux, “The agile manifesto, design thinking and systems engineering,” in *2017 Annual IEEE International Systems Conference (SysCon)*, pp. 1–5, 2017. https://doi.org/10.1109/SYSCON.2017.7934765.

[176] S. E. Carpenter and A. Dagnino, “Is agile too fragile for space-based systems engineering?,” in *2014 IEEE International Conference on Space Mission Challenges for Information Technology*, pp. 38–45, 2014. https://doi.org/10.1109/SMC-IT.2014.13.

[177] M. Warcholinski, “Differences between lean, agile and scrum,” 2019. https://brainhub.eu/blog/differences-lean-agile-scrum/, Accessed 2019-07-04.

[178] D. Selva, B. Dingwall, and S. Altunc, “A concept for an agile mission development facility for cubesat and suborbital missions,” in *IEEE Aerospace Conference Proceedings*, 2016. https://doi.org/10.1109/AERO.2016.7500564.

[179] K. Könnöla, S. Suomi, T. Mäkilä, and T. Lehtonen, “Can embedded space system development benefit from agile practices?,” *EURASIP Journal on Embedded Systems*, vol. 1, pp. 1–16, 2017. https://doi.org/10.1186/s13639-016-0040-z.

[180] J. Campos and P. Ferguson, “Manitobasat-1: Space systems engineering for student training,” in *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1–4, 2020.

[181] Y. Lee, A. Waterman, H. Cook, B. Zimmer, B. Keller, A. Puggelli, J. Kwak, R. Jevtic, S. Bailey, M. Blagojevic, P. F. Chiu, R. Avizienis, B. Richards, J. Bachrach, D. Patterson, E. Alon, B. Nikolic, and K. Asanovic, "An agile approach to building risc-v microprocessors," *IEEE Micro*, vol. 36, no. 2, pp. 8–20, 2016. `https://doi.org/10.1109/MM.2016.11`.

[182] A. I. Böhmer, P. Hugger, and U. Lindemann, "Scrum within hardware development insights of the application of scrum for the development of a passive exoskeleton," in *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pp. 790–798, 2017. `https://doi.org/10.1109/ICE.2017.8279965`.

[183] T. J. Mosher, J. Kolozs, and E. Wilder, "Agile hardware development approaches applied to space hardware," in *2018 AIAA SPACE and Astronautics Forum and Exposition*, 2018. `https://doi.org/10.2514/6.2018-5233`.

[184] H. Lykins and B. Cohen, "The INCOSE Model Driven System Design Interest Group," *INSIGHT: Model-Based Systems Engineering: A New Paradigm*, vol. 1, no. 3, 1998.

[185] INCOSE, *INSIGHT: Model-Based Systems Engineering: The New Paradigm*, vol. 12. THE INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING, 2009.

[186] C. C. Insaurralde, M. A. Seminario, J. F. Jimenez, and J. M. Giron-Sierra, "Model-based design analysis of an avionics fuel distributed control system," in *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, pp. 5.C.5–1–5.C.5–12, 2008.

[187] S. C. Spangelo, D. Kaslow, C. Delp, B. Cole, L. Anderson, E. Fosse, B. S. Gilbert, L. Hartman, T. Kahn, and J. Cutler, "Applying model based systems engineering (MBSE) to a standard CubeSat," in *IEEE Aerospace Conference Proceedings*, 2012. `https://doi.org/10.1109/AERO.2012.6187339`.

[188] S. C. Spangelo, J. Cutler, L. Anderson, E. Fosse, L. Cheng, R. Yntema, M. Bajaj, C. Delp, B. Cole, G. Soremekum, and D. Kaslow, "Model based systems engineering (MBSE) applied to Radio Aurora Explorer (RAX) CubeSat mission operational scenarios," in *IEEE Aerospace Conference Proceedings*, 2013. `https://doi.org/10.1109/AERO.2013.6496894`.

[189] D. Kaslow, G. Soremekun, H. Kim, and S. Spangelo, "Integrated model-based systems engineering (MBSE) applied to the Simulation of a CubeSat mission," in *IEEE Aerospace Conference Proceedings*, 2014. `https://doi.org/10.1109/AERO.2014.6836317`.

[190] A. I. Anyanhun and W. W. Edmonson, "An mbse conceptual design phase model for inter-satellite communication," in *2018 Annual IEEE International Systems Conference (SysCon)*, pp. 1–8, 2018.

[191] J. Holt, S. Perry, R. Payne, J. Bryans, S. Hallerstede, and F. O. Hansen, "A model-based approach for requirements engineering for systems of systems," *IEEE Systems Journal*, vol. 9, no. 1, pp. 252–262, 2015. `https://doi.org/10.1109/JSYST.2014.2312051`.

[192] D. R. Wibben and R. Furfaro, "Model-based systems engineering approach for the development of the science processing and operations center of the nasa osiris-rex asteroid sample return mission," *Acta Astronautica*, vol. 115, pp. 147–159, 2015.

[193] S. Gebreyohannes, W. Edmonson, and A. Esterline, "Formal behavioral requirements management," *IEEE Systems Journal*, vol. 12, no. 3, pp. 3006–3017, 2018.

[194] R. d. Ferluc, A. Provost-Grellier, and B. Dellandrea, "Model based fdir process with capella and compass," in *MBSE ESA Workshop*, 2016.

[195] J. Evans, S. Cornford, and M. S. Feather, "Model based mission assurance: Nasa's assurance future," in *2016 Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–7, 2016.

[196] G. Biggs, A. Armonas, T. Juknevicius, and K. Post, "Integrating safety and reliability analysis into mbse: overview of the new proposed omg standard," in *INCOSE International Symposium*, vol. 28, pp. 1322–1336, 2018. https://doi.org/10.1002/j.2334-5837.2018.00551.x.

[197] R. Kratzke, "Failure modes effects analysis in mbse," in *Texas Gulf Coast Chapter*, Series Failure Modes Effects Analysis in MBSE, (Houston, TX), 2018.

[198] R. S. Kalawsky, J. O. Brien, S. Chong, C. Wong, H. Jia, H. Pan, and P. R. Moore, "Bridging the gaps in a model-based system engineering workflow by encompassing hardware-in-the-loop simulation," *IEEE Systems Journal*, vol. 7, no. 4, pp. 593–605, 2013.

[199] B. Nastov, V. Chapurlat, F. Pfister, and C. Dony, "Mbse and v&v: a tool-equipped method for combining various v&v strategies," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10538–10543, 2017.

[200] E. A. Bjorkman, S. Sarkani, and T. A. Mazzuchi, "Using model-based systems engineering as a framework for improving test and evaluation activities," *Systems Engineering*, vol. 16, no. 3, pp. 346–362, 2013.

[201] S. Piggott, P. Melanson, and L. Hartman, "A vision for super-model driven systems engineering," in *INCOSE International Symposium*, vol. 17, pp. 52–65, 2007.

[202] J.-L. Voirin, S. Bonnet, V. Normand, and D. Exertier, "From initial investigations up to large-scale rollout of an mbse method and its supporting workbench: the thales experience," *INCOSE International Symposium*, vol. 25, no. 1, pp. 325–340, 2015.

[203] C. Haskins, "4.6.1 a historical perspective of mbse with a view to the future," in *INCOSE International Symposium*, vol. 21, pp. 493–509, 2011.

[204] T. Huldt and I. Stenius, "State-of-practice survey of model-based systems engineering," *Systems Engineering*, vol. 22, no. 2, pp. 134–145, 2019. https://doi.org/10.1002/sys.21466.

[205] K. Henderson and A. Salado, "Value and benefits of model-based systems engineering (MBSE): Evidence from the literature," *Systems Engineering*, vol. 24, no. 1, pp. 51–66, 2020. https://doi.org/10.1002/sys.21566.

[206] L. Anderson, B. Cole, R. Yntema, M. Bajaj, S. Spangelo, D. Kaslow, C. Lowe, E. Sudano, M. Boghosian, R. Reil, S. Asundi, and S. Friedenthal, "Enterprise modeling for cubesats," in *IEEE Aerospace Conference Proceedings*, 2014. https://doi.org/10.1109/AERO.2014.6836343.

[207] D. Kaslow, L. Anderson, S. Asundi, B. Ayres, C. Iwata, B. Shiotani, and R. Thompson, "Developing a CubeSat Model-Based System Engineering (MBSE) Reference Model -

Interim status," in *IEEE Aerospace Conference Proceedings*, vol. 2015-June, 2015. https://doi.org/10.1109/AERO.2017.7943691.

[208] D. Kaslow, B. J. Ayres, P. T. Cahill, M. J. Chonoles, L. Hart, C. Iwata, A. G. Levi, and R. Yntema, "CubeSat Model-Based System Engineering (MBSE) reference model - Development and distribution - Interim status," in *AIAA Space and Astronautics Forum and Exposition, SPACE 2016*, 2016. https://doi.org/10.2514/6.2016-5551.

[209] D. Kaslow, L. Hart, B. Ayres, C. Massa, M. J. Chonoles, R. Yntema, B. Shiotani, and S. D. Gasster, "Developing a CubeSat Model-Based System Engineering (MBSE) Reference Model – Interim Status #2," in *IEEE Aerospace Conference*, (Big Sky, MT, USA), pp. 1–16, IEEE, 2016. https://doi.org/10.1109/AERO.2016.7500592.

[210] D. Kaslow, "CubeSat Model-Based Systems Engineering (MBSE) Reference Model - Development and Distribution – Interim Status #3," in *CubeSat Workshop*, 2017.

[211] D. Kaslow, L. Hart, B. Ayres, P. T. Cahill, and R. Yntema, "A Model-Based Systems Engineering (MBSE) Approach for Defining the Behaviors of CubeSats," in *IEEE Aerospace Conference*, 2017. https://doi.org/10.1109/AERO.2017.7943865.

[212] D. Kaslow, P. T. Cahill, and R. Frank, "Developing a CubeSat System MBSE Reference Model - Interim Status #5," in *Proceedings of the AIAA/USU Conference on Small Satellites*, (Logan, UT), 2019. SSC19-P2-16, https://digitalcommons.usu.edu/smallsat/2019/all2019/200/.

[213] A. Alanazi and J. Straub, "Engineering methodology for student-driven cubesats," *Aerospace*, vol. 6, no. 5, 2019. https://doi.org/10.3390/aerospace6050054.

[214] J. M. Gregory, J. S. Kang, R. Bruninga, R. M. Sega, J. M. Borky, D. Kaslow, and G. H. Downey, "Applying a Model-Based Approach to Develop a Standardized Template for CubeSat-class Satellites," in *IEEE Aerospace Conference Proceedings*, 2020. https://doi.org/10.1109/AERO47225.2020.9172748.

[215] J. Ring and A. M. Madni, "Key challenges and opportunities in 'system of systems' engineering," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 973–978 Vol. 1, 2005.

[216] A. Madni and M. Sievers, "Model-based systems engineering: Motivation, current status, and needed advances," in *Disciplinary Convergence in Systems Engineering Research*, pp. 311–325, John Wiley & Sons, Inc., 2018.

[217] R. A. Noguchi, R. J. Minnichelli, and M. J. Wheaton, "Architecting success in model based systems engineering pilot projects," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 755–760, 2019.

[218] B. Cameron and D. M. Adsit, "Model-based systems engineering uptake in engineering practice," *IEEE Transactions on Engineering Management*, vol. 67, no. 1, pp. 152–162, 2020. https://doi.org/10.1109/TEM.2018.2863041.

[219] T. S. Kuhn, *The Structure of Scientific Revolutions*. Chicago 60627: The University of Chicago Press, 2 ed., 1970.

[220] T. A. Schwandt, *The SAGE Dictionary of Qualitative Inquiry*. Thousand Oaks, California: SAGE Publications Inc., 3 ed., 2007.

[221] W. P. S. Dias, "Philosophical underpinning for systems thinking," *Interdisciplinary Science Reviews*, vol. 33, no. 3, pp. 202–213, 2008. https://doi.org/10.1179/174327908X366897.

[222] R. D. Arnold and J. P. Wade, "A complete set of systems thinking skills," *INSIGHT*, vol. 20, no. 3, pp. 9–17, 2017.

[223] E. Caillaud, B. Rose, and V. Goepp, "Research methodology for systems engineering: some recommendations," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1567–1572, 2016. https://doi.org/10.1016/j.ifacol.2016.07.803.

[224] P. Reason and H. Bradbury, eds., *The SAGE Handbook of Action Research*. SAGE Publications, Inc., 2nd ed., 2008.

[225] H. A. Hahn, "Planning for action research: Looking at practice through a different lens," in *Conference: 16th World Multi-Conference on Systemics, Cybernetics and Informatics*, 8 2012.

[226] M. C. Edson, "Team systems theory," in *Handbook of Systems Sciences* (G. S. Metcalf, K. Kijima, and H. Deguchi, eds.), pp. 1–44, Singapore: Springer Singapore, 2020.

[227] W. F. Whyte, ed., *Participatory Action Research*. Thousand Oaks, California: SAGE Publications, Inc., 1991.

[228] D. Tripp, "Action research: a methodological introduction." https://researchrepository.murdoch.edu.au/id/eprint/34154/1/actionresearch.pdf.

[229] H. B. Huang, "What is good action research?," *Action Research*, vol. Vol. 8, no. No. 1, pp. 93–109, 2010. https://doi.org/10.1177/1476750310362435.

[230] S. E. Hove and B. Anda, "Experiences from conducting semi-structured interviews in empirical software engineering research," in *11th IEEE International Software Metrics Symposium (METRICS'05)*, pp. 10 pp.–23, 2005. https://doi.org/10.1109/METRICS.2005.24.

[231] R. K. Yin, *Case Study Research and Applications: Design and Methods*. SAGE Publications Inc., 6 ed., 2018.

[232] P. Lavrakas, *Encyclopedia of Survey Research Methods*, vol. 1. Thousand Oaks, California: Sage Publishing, 2008.

[233] H. N. Boone and D. A. Boone, "Analyzing likert data," *Journal of extension*, vol. 50, no. 2, pp. 1–5, 2012.

[234] J. H. Zar, "Spearman rank correlation," *Encyclopedia of biostatistics*, vol. 7, 2005.

[235] R. S. Nickerson, "Confirmation bias: A ubiquitous phenomenon in many guises," *Review of General Psychology*, vol. 2, no. 2, pp. 175–220, 1998. https://doi.org/10.1037/1089-2680.2.2.175.

[236] "Replication," 2021/12/07 2010.

[237] J. Elstak, R. Amini, and R. Hamann, "A comparative analysis of project management and systems engineering techniques in cubesat projects," in *19th Annual International Symposium of the International Council on Systems Engineering, INCOSE 2009*, vol. 2, pp. 545–559, 2009.

[238] Administrator, "Delfi-c3 - project sponsors." Accessed 2021-12-13, `https://web.archive.org/web/20110927174248/http://www.delfic3.nl/index.php?option=com_content&task=view&id=45&Itemid=83`.

[239] Star Lab, "Iris cubesat," 2021. Accessed 2021-12-09, `https://sites.google.com/site/manitobasat1/home?authuser=0`.

[240] N. Garzaniti and A. Golkar, "Performance assessment of agile hardware co-development process," in *2020 IEEE International Symposium on Systems Engineering (ISSE)*, pp. 1–6, Year of Conference.

[241] R. Lyells, R. Angier, and B. Epps, "Creating decision guidance for applying agile system engineering," in *INCOSE International Symposium*, vol. 28, pp. 655–670, 2018. `https://doi.org/10.1002/j.2334-5837.2018.00507.x`.

[242] B. Sauser and J. Boardman, "Systemigram modeling for contextualizing complexity in system of systems," in *Modeling and Simulation Support for System of Systems Engineering Applications* (L. B. Rainey and A. Tolk, eds.), John Wiley and Sons, 2015.

[243] P. Davies, *Don't Panic - The Absolute Beginner's Guide to Managing Interfaces*. Great Britain: INCOSE UK Limited, 2019.

[244] Q. Wu, D. Gouyon, E. Levrat, and S. Boudau, "Use of patterns for know-how reuse in a model-based systems engineering framework," *IEEE Systems Journal*, vol. 14, no. 4, pp. 4765–4776, 2020. `https://doi.org/10.1109/JSYST.2020.2975116`.

[245] T. Katz, "Evaluation of cots hardware assemblies for use in risk averse, cost constrained space-based systems," *INCOSE International Symposium*, vol. 29, no. 1, pp. 866–878, 2019.

[246] Z. Tompkins, M. Grenn, and B. Roberts, "Improving system maturity assessments by incorporating a design structure matrix," *IEEE Transactions on Engineering Management*, vol. 67, no. 1, pp. 122–140, 2020. `https://doi.org/10.1109/TEM.2018.2867440`.

[247] B. Sauser, J. Ramirez-Marquez, R. Magnaye, and W. Tan, "A systems approach to expanding the technology readiness level within defense acquisition," tech. rep., Stevens Institute of Technology, 03/20 2009.

[248] A. L. Olechowski, S. D. Eppinger, N. Joglekar, and K. Tomaschek, "Technology readiness levels: Shortcomings and improvement opportunities," *Systems Engineering*, vol. 23, no. 4, pp. 395–408, 2020. `https://doi.org/10.1002/sys.21533`.

[249] K. Bakke and C. Haskins, "Use of trl in the systems engineering toolbox," *INCOSE International Symposium*, vol. 28, no. 1, pp. 587–601, 2018.

[250] R. Gove, *Development of an integration ontology for system operational effectiveness*. PhD thesis, Stevens Institute of Technology, 2007.

[251] B. J. Sauser, J. E. Ramirez-Marquez, D. Henry, and D. DiMarzio, "A system maturity index for the systems engineering life cycle," *International Journal of Industrial and Systems*

Engineering, vol. 3, no. 6, pp. 673–691, 2008.

[252] B. J. Sauser, M. Long, E. Forbes, and S. E. McGrory, "3.1.1 defining an integration readiness level for defense acquisition," *INCOSE International Symposium*, vol. 19, no. 1, pp. 352–367, 2009.

[253] J. Mankins, "Technology readiness level," tech. rep., Office of Space Access and Technology, NASA, 1995.

[254] M. Knaggs, D. Harkreader, A. Unione, J. Oelfke, J. Ramsey, D. Keaims, B. Sauser, and B. Atwater, "Nesting in the evaluation of system readiness for complex systems of emerging technologies," in *2017 Annual IEEE International Systems Conference (SysCon)*, pp. 1–6, 2017. `https://doi.org/10.1109/SYSCON.2017.7934811`.

[255] E. Kujawski, "Analysis and critique of the system readiness level," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 979–987, 2013. `https://doi.org/10.1109/TSMCA.2012.2209868`.

[256] J. Robinson, D. Levack, R. Rhodes, and T. Chen, "Need for technology maturity of any advanced capability to achieve better life cycle cost," in *45th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, AIAA, 2009.

[257] M. Miller, S. Thomas, and C. Rusnock, "Extending system readiness levels to assess and communicate human readiness," *Systems Engineering*, vol. 19, no. 2, pp. 146–157, 2016. `https://doi.org/10.1002/sys.21344`.

[258] C. Haskins, *Systems engineering analyzed, synthesized, and applied to sustainable industrial park development*. Thesis, NTNU, 2008.

[259] A. J. Shenhar, V. Holzmann, B. Melamed, and Y. Zhao, "The challenge of innovation in highly complex projects: What can we learn from boeing's dreamliner experience?," *Project Management Journal*, vol. 47, no. 2, pp. 62–78, 2016. `https://doi.org/10.1002/pmj.21579`.

[260] M. W. Potts, A. Johnson, and S. Bullock, "Evaluating the complexity of engineered systems: A framework informed by a user case study," *Systems Engineering*, vol. 23, no. 6, pp. 707–723, 2020. `https://doi.org/10.1002/sys.21558`.

[261] G. Elia, A. Margherita, and G. Secundo, "Project management canvas: a systems thinking framework to address project complexity," *International Journal of Managing Projects in Business*, vol. 14, no. 4, pp. 809–835, 2021. `https://doi.org/10.1108/IJMPB-04-2020-0128`.

[262] J. Geraldi, H. Maylor, and T. Williams, "Now, let's make it really complex (complicated)," *International Journal of Operations & Production Management*, vol. 31, no. 9, pp. 966–990, 2011. `https://doi.org/10.1108/01443571111165848`.

[263] S. Bakken, E. Honoré-Livermore, R. Birkeland, M. Orlandic, E. F. Prentice, J. L. Garrett, D. D. Langer, C. Haskins, and T. A. Johansen, "Software Development and Integration of a Hyperspectral Imaging Payload for HYPSO-1," in *2022 IEEE/SICE International Symposium on System Integration (SII)*, 2021. Accepted.

[264] J. Zhang, C. Haskins, Y. Liu, and M. A. Lundteigen, "A systems engineering–based approach for framing reliability, availability, and maintainability: A case study for subsea design," *Systems Engineering*, vol. 21, no. 6, pp. 576–592, 2018. `https://doi.org/10.1002/sys.21462`.

[265] N. J. Lindsey, M. Alimardani, and L. D. Gallo, "Reliability analysis of complex nasa systems with model-based engineering," in *66th Annual Reliability & Maintainability Symposium (RAMS)*, (Palm Springs, CA, United States), 2020.

[266] E. Brusa, "Digital twin: Towards the integration between system design and rams assessment through the model-based systems engineering," *IEEE Systems Journal*, pp. 1–12, 2020. `https://doi.org/10.1109/JSYST.2020.3010379`.

[267] A. Sols and A. Salado, "Integrating reliability in systems engineering management," *Engineering Management Journal*, vol. 31, no. 3, pp. 207–221, 2019. `https://doi.org/10.1080/10429247.2019.1632664`.

[268] F. Elakramine, R. Jaradat, N. U. I. Hossain, M. Banghart, C. Kerr, and S. E. Amrani, "Applying systems modeling language in an aviation maintenance system," *IEEE Transactions on Engineering Management*, pp. 1–13, 2021. `https://doi.org/10.1109/TEM.2021.3089438`.

[269] A. K. O. Moen, E. Sjoevold, and O. Jordheim, *Implementation of FMECA in Small Satellite Development*. Bachelor thesis, Norwegian University of Science and Technology, 2019.

[270] P. Svenson and J. Axelsson, "Should I Stay or Should I Go? How Constituent Systems Decide to Join or Leave Constellations in Collaborative SoS," in *2021 16th International Conference of System of Systems Engineering (SoSE)*, pp. 179–184, Year of Conference. `https://doi.org/10.1109/SOSE52739.2021.9497474`.

[271] B. W. Oppenheim and C. Haskins, "Powerful opportunities to improve program performance using lean systems engineering and lean program management," in *Advances in Systems Engineering* (R. C. John Hsu, ed.), vol. 252, Reston, Virginia, USA: American Institute of Aeronautics and Astronautics, 2016.

[272] E. Fosse, "Model-based systems engineering mbse 101," 2016. Presented at the INCOSE International Workshop 2016.

[273] A. Bryman, *Social research methods*. New York, USA: Oxford University Press, 2012.

[274] N. Norris, "Error, bias and validity in qualitative research," *Educational Action Research*, vol. 5, no. 1, pp. 172–176, 1997.

Chapter 8

Papers

This portion of the thesis provides the publications in the order listed in Chapter 1. These papers present the main contributions for addressing the research objective and questions.

Paper A

Honoré-Livermore, Evelyn, "CubeSats in University: Using Systems Engineering Tools to Improve Reviews and Knowledge Management," Conference on Systems Engineering Research, *Procedia Computer Science*, Volume 153, pp. 63–70, 2019 [12]. `https://doi.org/10.1016/j.procs.2019.05.056`.

Here, the problem space of developing CubeSats at universities is described, as well as some of the specific challenges experienced in the HYPSO team.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 153 (2019) 63–70

Procedia Computer Science

www.elsevier.com/locate/procedia

17th Annual Conference on Systems Engineering Research (CSER)

CubeSats in University: Using Systems Engineering Tools to Improve Reviews and Knowledge Management

Evelyn Honoré-Livermore[a,*]

[a]*Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway*

Abstract

Coordinating research objectives concurrently with product development and engineering is a challenge in student-run CubeSat projects. The literature review from several universities and the International Council of Systems Engineering (INCOSE) Space Systems Working Group (SSWG) all acknowledge the need for a better methodology. This paper describes findings from a university CubeSat exploratory case study related to team management, reviews and design. Introducing Systems Engineering (SE) tools such as formalized reviews, A3 reports, systemigrams and N2 dependency maps have aided the development process. This was evident through faster development iterations and fewer design confusions. Results show that relocation into a common space based on recommendations from Concurrent Engineering (CE) has improved overall communication and team work.

Keywords: CubeSat; knowledge management, systems engineering, concurrent engineering, systemigram, reviews

1. Introduction and Background

The CubeSat standard for small satellites has enabled faster implementation and engineering of satellites that can be used for both research activities and educational purposes. PhD-level candidates now find themselves a part of CubeSat design groups and are asked to contribute to engineering activities for designing and building the satellites.

* Corresponding author. Tel.: +47 400 18 398;
E-mail address: evelyn.livermore@ntnu.no

1877-0509
10.1016/j.procs.2019.05.056

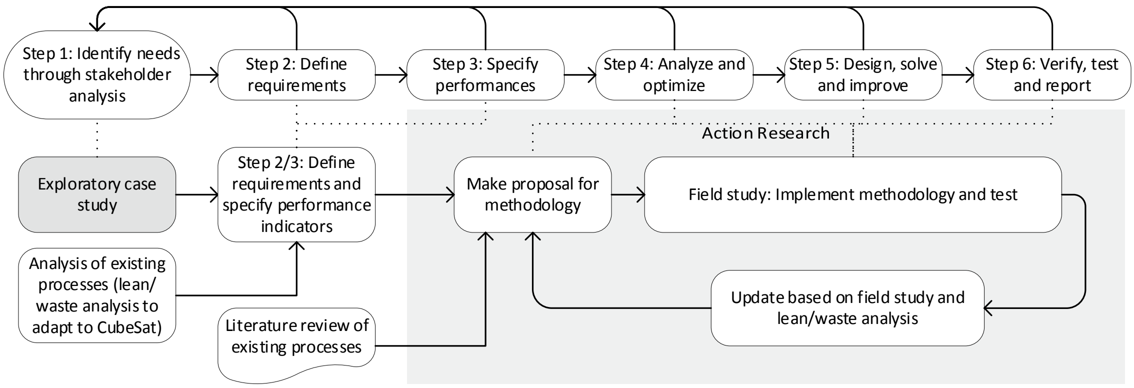


Fig. 1. The implementation of the Systems Engineering (SE) process [1] for the author's research project is based on the work by Sopha et. al [2] (top row). Dashed lines show the author's process' relationship to the original process. From the exploratory case study of HYPer-spectral Smallsat for ocean Observation (HYPSO) (left grey box) requirements and performance indicators are established (e.g. team engagement, schedule, cost). The literature review and analysis of existing processes relevant to CubeSats combined with a case study will result in a proposed methodology. This will be tested in future case study and updated to improve the performance indicators if necessary. E.g. lean analysis can be used to indicate how to improve.

Although this is exciting and motivating, the individual research goals they should pursue may suffer under the load of engineering activities. They then find themselves either demotivated to do research activities because they find engineering satellites more engaging, or they leave the CubeSat project team abruptly and the engineering project suffers.

Methods to ensure fulfillment of both activities, ensuring good research and good engineering, is important at universities where PhD candidates participate in satellite projects – or where the payload data of a satellite is relevant to the research. Furthermore, ensuring that proper research is carried out is difficult in a fast-developing field, where what was considered research last year is considered mass production and engineering today.

This exploratory case study presents some features of a CubeSat project at the Norwegian University of Science and Technology (NTNU), to be used as a basis for the overall research project aimed at improving the process shown in Fig. 1 for developing CubeSats in a university setting. This paper focuses on findings from the first year of case study, where challenges of team management have required the most effort.

1.1. CubeSats at Universities

CubeSats are the typical university-built satellites. The CubeSat is a standardized format and there is a multitude of resources available, including Commercial-Off-The-Shelf (COTS) components and several commercial suppliers of turnkey systems. The projects at universities range from fully in-house developed CubeSats; to payload development and integration to a standard purchased bus; combination of COTS CubeSat subsystems with in-house developed subsystems; to mission control and operations of other satellites. According to NanoSat, academia stands for ~30% of the CubeSats launched the past years, close to 150 in 2018 [3]. NTNU has a history of space and CubeSat related activities [4] where a combination of student-led activities and course subjects have led to the generation of several MSc theses and credits, but with no successful missions to date.

There are several challenges when conducting engineering projects at universities; project management and team structure [5], [6], [7], [8]; Birkeland et al. [9] discuss how one of the main issues are the ever-changing teams and short time for on-boarding new members; balancing coursework and satellite building [8]; ensuring momentum [6], [7]; and ensuring success of mission [5], [6]. In summary, the studies have recommended: (1) Ensure that funding is secured; (2) Interface control is key to make a multidisciplinary product team successful; (3) Give ownership to the students. Integrate them as a team and have continuous team building activities; (4) Schedule the project so that exam periods and holidays are respected; (5) Establish a process for knowledge transfer; and finally, (6) Manage expectations of stakeholders: students and supervisors and external players.

1.2. Systems Engineering and Sociotechnical Research

Systems Engineering (SE) has been around since the beginning of the 1900s and is the basis of how we do complex product development and life cycle management in many industries today. Through taking a holistic view of product development, its tools and processes are relevant to university-managed projects. *Sociotechnical systems* is a soft SE area that considers how people and organizations behave and act in the project context [10].

The sociotechnical viewpoint is relevant in a university setting where there is a high turnover of people, because it is difficult to build a specific organization culture that everyone in the team adapts quickly. The organizational culture should be like the cultural setting these teams experience to make the transition and adaptation quicker.

Through the application of SE tools and processes and using models to depict and communicate the sociotechnical systems, we attempt to develop a methodology that will enable better fulfillment of both research goals and engineering goals in a university setting. This paper is based on the first year of experiences, where much effort has been spent to understand the needs of the project and the context. Some events have led to improvement of processes through tools from SE.

1.3. Method: Case Study Research

Case study research, defined by Bromley [11] as "…a systematic inquiry into an event or a set of related events which aims to describe and explain the phenomenon of interest", is utilized in this research through an *exploratory case study* [12]. This may be elaborated into explanatory or descriptive case studies once the research problem becomes further defined.

The exploratory case study's strength is the inherent characteristic of not knowing what to look for – enabling discovery of unexpected phenomena [13]. It is also recommended by Caillaud et al. [14] as a method for SE research. Both Yin [13] and Caillaud et al. [14] highlight how exploratory case studies have parallels to grounded theory work, where the goal is to "lead(ing) to a theory" [14] – which is the overall objective of the research project, as shown in Fig. 1. The criticism is based on the nature of exploratory case: "…lack of specific, theory-based prior assumptions are often not considered a strength but a weakness. There is, of course, always the risk that these characteristics could be an excuse for inadequate and unscientific studies." [14]

The background for choosing the method of case study research as opposed to other social science methods is that the NTNU CubeSat project is new, and there is no previous experience nor evidence for where the strengths or weaknesses lie. An exploratory case study approach offers a broad perspective to understand the project's characteristics and looking for methods to improve the success of both academic research and the engineering project.

2. Case Study: HYPer-spectral Smallsat for ocean Observation (HYPSO)

The purpose of the case study is to understand how research projects are conducted influences their success, and how engineering and academic research can be combined while achieving goals in both fields, and to extract some best practices or lessons learned. The project organization to help achieve the goals is the phenomenon of interest through modes of interaction and communication, and the results relevant to similar sociocultural contexts.

The following research questions are addressed:

- RQ-1: How can an engineering project ensure the fulfillment of academic research goals in a university setting?
- RQ-2: How can engineering goals and individual research goals be fulfilled simultaneously?
- RQ-3: What methods and modes of interactions in a university research project are present, work well, and why?

The author has a background from *old-space*, where one is required to follow a set of processes and standards, and the organization is built around this. With this perspective, the author will be biased and often notice the differences and may view the less organized *new-space* negatively when compared to the traditional methodology.

HYPer-spectral Smallsat for ocean Observation (HYPSO) is a satellite mission funded by Research Council of Norway (RCN), Centre of Autonomous Marine Operations and Systems (AMOS) and supported by the Departments of Electronic Systems and Engineering Cybernetics at NTNU. The HYPSO project team today consists of approximately 20 students and 10 PhD-level employees. Furthermore, there are about 10 professors closely or loosely associated with the project, which have different needs and expectations to the project and its execution. One of the challenges with the HYPSO project team is that few or none have experience in designing and building CubeSats, neither the professors nor the students. The goal of the satellite is to support oceanographic studies as funded by RCN; build competence in CubeSat design and manufacturing at NTNU and experience in team work for students.

Project history: Nov 2016: Pre-project deadline. At the pre-project deadline, an application was submitted to RCN fund the cost of defining and describing the MASSIVE (Mission-oriented autonomous systems with small satellites for maritime sensing, surveillance and communication) project. People involved: mainly professors interested in the outcome of the satellite data. **Apr 2017: Pre-project start.** The pre-project consisted of defining the project in detail, and describing the Scope of Work, Cost, and Schedule. Many trade-offs for system architecture performed using Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) and Analytic Hierarchy Process (AHP). People involved: professors, scientists, mission architect (PhD candidate), remote sensing PhD, third PhD working on holistic view with satellites. **Nov 2017: Additions to team.** Two PhD candidates were hired with the goal of supporting the MASSIVE project, but not funded by the project itself. **Dec 2017: Mission Design Review (MDR).** Review process and evaluation described in Section 4.1. **Jan 2018: Project granted, and kick-off held.** Full project was granted by RCN. A kick-off involving the five PhD candidates, and a MSc student associated with the project was held, trying to agree on how to work together and what to focus on. **Mar 2018: Disruption in team.** One of the five key members left the project on secondment: still available via email. **August 2018: Project kick-off.** After recruiting ~20 MSc/BSc students, the project team had a new kick-off with a total of almost 30 people.

3. Analysis and Discussion

This section covers first a description and analysis of the review process, secondly improvements in the design process and knowledge management are discussed.

In the brief history of the project, two formal reviews have been conducted. The first, MDR, was held in Dec. 2017, without a strict process. The second, Preliminary Design Review (PDR), was held in Oct. 2018, following the reduced European Cooperation for Space Standardization (ECSS) process of reviews. The reduced process has been tailored to CubeSats and other new-space applications. An overview of the two review processes, as well as the baseline ECSS process is given in Fig. 2.

3.1. Review Process

Mission Design Review: While the preparation to MDR did not follow a specific format, the relevant people were invited, and the agenda was clear from the invitation. Some participants of the review team had been a part of the project from the start while others joined the project 2 months prior to MDR. The data package was provided the day before the review, consisting of only a presentation covering the mission design. The late sharing of the review item gave the review team little time to prepare. During the meeting, the work was presented, and there were several discussions about the various topics. The form provided at the end of the meeting gave the review team an opportunity to give written feedback which impacted some design decisions. There was no clear traceability of these decisions to the requirements. Two trade-off analyses were conducted, on satellite size and camera size. An Orbit Analysis Report would have been helpful at this stage in the project, as well as the Mission Requirements Document, but these were not generated until later.

The feedback from the review team on the review itself highlighted the following issues: (1) little time to prepare for the meeting; (2) difficult to give feedback with so much information presented at the meeting; (3) no clear tracing of feedback being implemented. Furthermore, the project team itself felt that it did not gain ownership of the mission through the review or the subsequent work, which has become apparent at several occasions in the design process and daily work. It would also have been helpful to de-scope the mission at this stage, which had grown the past half year. In summary, it was more an academic-inspired review than a formal engineering review.

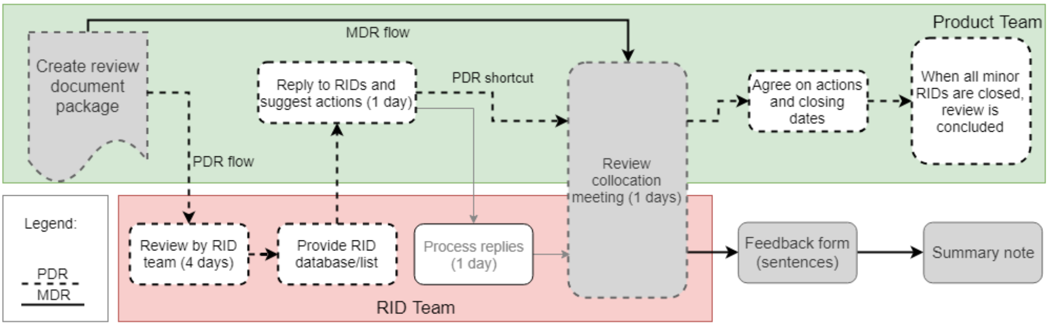


Fig. 2. The Review Process based on ECSS, where “Feedback from (sentences) and “Summary note” boxes are not part of original ECSS. MDR flow: Grey boxes and solid flow lines. PDR flow: boxes with dashed frames and dashed flow lines. The box “Process replies” was not performed.

Preliminary Design Review: The participants were invited 3-4 months beforehand. A formal invitation with a procedure based on ECSS was sent to the review team 1-2 months prior to the collocation meeting, stating the process and objectives of the review. The data package was provided a week prior to the review, and the Review Item Discrepancies (RIDs) were recorded in a format that allowed for easier follow-up of comments. The review team provided feedback on time, and the project team responded to the RIDs. However, there was not enough time to let the review team look at the replies because the project team spent more time than expected providing sufficiently detailed responses. At the meeting, the RIDs were processed per document. Each RID was presented, and the feedback from the project team given. A disposition and an action were agreed on, with a deadline for completing the action.

The feedback from review and project team included: (1) the RID form helped in providing feedback; (2) too little time to review; (3) clearer description of mission/be invited to a mission review; (4) poor traceability of spacecraft requirements; (5) better structuring and overview of documentation would have helped when reviewing; (6) two day collocation (one day to go through RIDs and one day to work on technical discussions/workshops).

The project team felt that making the documentation and having a clear deadline helped moving forward with the design. The project team worked together on a set of documents on Google Drive which allowed for concurrent editing and writing, as well as simple interface to assign tasks and ask questions about design.

3.2. Knowledge Management and Design Process

Modern-day trends and lean theory support the use of non-serial processes such as agile methodology or SPADE [15]. Clegg and Boardman [16] compare three different philosophies aiming to cope with non-serial product development lifecycle: Business Process Reengineering (BPR), Soft Systems Methodology (SSM), and Concurrent Engineering (CE). In this paper, the argument for considering SSM is that “human nature can be erratic, illogical and unpredictable, (and) the system can become difficult to define” [17]. Humans are much more difficult to study than physical systems [18]. SSM recommends using modelling tools such as systemigrams to highlight the actors and thus the human nature in the system. The systemigram is useful because it can show how inputs and outputs are transformed at different interfaces, the purpose of the system, the role of each actor in the system ([16]-[18]), but at the same time it can contain a lot of information which may be confusing to the reader.

The project structure is represented by the systemigram in Fig. 3. Developing the systemigram for the HYPSO project has clarified some of the unknown interactions and patterns in the system and uncovering the complexity in managing the expectations of different stakeholders. The systemigram shows how the *Mission Objectives* have been developed by the *Project stakeholders,* while it is common in university CubeSat projects that this is a stronger collaborative effort together with the *Project execution*. This has been identified as one of the main reasons for PhD candidates losing interest. The mission requirements were updated to include *“achieve research goals”*, but it was not followed up with actual requirements or actions with deadlines.

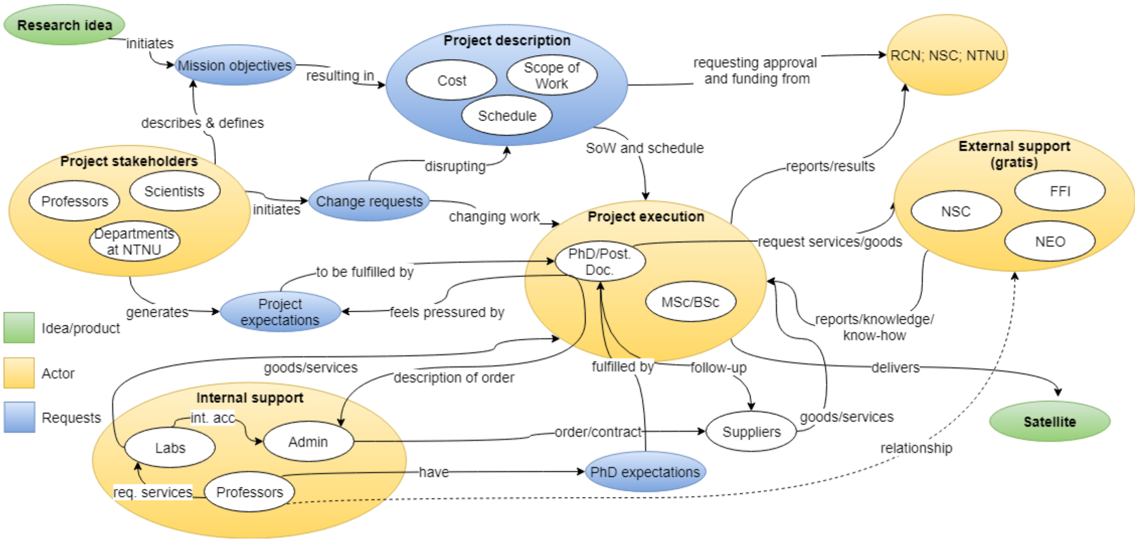


Fig. 3. Project Systemigram. The *Project stakeholders* are looking for higher performance and faster product delivery, the *Internal support* are interested in seeing the *Project execution* team succeed with both the theses and the project goals (at the same time), and the funding bodies *RCN, Norwegian Space Agency (NSC),* and *NTNU* want everything to be completed and published on time and on cost. The *Project execution* wants both to fulfil the expectations for their theses to finish their formal tasks on time to finish their degrees, and to build a satellite that satisfies the *Project stakeholders*.

A secondary payload with a corresponding secondary mission has been included in the satellite since December 2018. For this mission, the objectives have been derived directly from the PhD research objectives. The payload itself is spaceflight proven, but the software and mission design are new. The future reporting of the case study will include analysis of this secondary mission and the effects on motivation and progress of research and engineering project.

Prior to August 2018 kick-off: The project team consisted of 5 PhDs, 1 Post.Doc. and 5-6 MSc loosely associated to the project. The MSc students were encouraged to sit together to work but were working on separate blocks of the project. The rest of the project team were in different offices spread over a building. The continuous project work consisted of having weekly project meetings and trying to agree on a way forward for the product development. However, because of other obligations (coursework, lack of time, duty work), not much progress was made. Some of the players made progress but did not have the time to share the knowledge in a way that made the project team understand and build on it well enough. Furthermore, the communication from the *Project stakeholders* was very strong and controlling, which may have made it difficult for the project team to feel ownership; resulting in a lack of motivation.

Another reason for lack of motivation was the distributed team structure. Relying on formal meetings for communication made the development iterations lengthy, and immediate issues might be forgotten between. The MSc team functioned much better in this aspect, maintaining a better line of communication and daily interactions.

Post August 2018 kick-off: The greatest change from pre/post August kick-off was the moving of team into the same working space. This is based on the CE principles, where collocation of project team has shown to have improved efficiency and increased both formal and informal communication [19]. The team now has set working times (3 times of 4-5 hours per week) and are always encouraged to be in the same office space.

The results so far have been promising, and this is evident through the continuous discussions and faster implementation of design choices. There are more informal interactions, laughter and open body language than previously observed. This can indicate that there is a higher degree of trust, and people who trust each other can make better design choices because they share information more readily. Additionally, being in the same room lowers the threshold for seeking information or for clarifying uncertainties and unknowns.

Some of the design confusions happened because disciplines made decisions that influenced other disciplines but did not know to inform each other. Most of the team members had no previous experience with working in multidisciplinary teams. To combat this, an exercise of jointly creating an N2 dependency map was conducted, bringing awareness of dependencies between the disciplines, and understanding of who to contact and inform if changes were made.

In the kick-off, it was decided that all documentation and working files should be on Google Drive, to allow for transparency and easy flow of information. After a month, it became clear that this was not enough. A3 reports [20] were introduced in the form of post-its on A3 sized paper on the wall.

Lastly, there have been several internal workshops on how to achieve individual research goals concurrently with engineering goals, and the inclusion of research goals in the mission requirements have aided in maintaining this dual focus. Joint efforts on paper brainstorming and scheduling the engineering goals so that individual research goals may be achieved has also been performed. To this date, it is not known whether these measures have been successful.

4. Conclusion

The case study has uncovered some benefits from usage of SE tools and methods, namely: formalized review process, concurrent engineering, A3 reports, N2 dependency mapping, and systemigram. The development of systemigram in Fig. 3 uncovered the mechanisms in how the project and mission had been established and clarified some reasons to lack of motivation and poor alignment between engineering goals and PhD research objectives. The systemigram has then aided the understanding of the project itself, which in turn can help in improving the project to answer the research questions.

RQ-1/2: Workshops and inclusion of more detailed research mission requirements were introduced to facilitate the focus on the duality of engineering and research. Because the project needs the PhDs for execution, there is still a potential for conflict. However, removing the PhDs from *Project execution* and allowing them to only focus on their research objectives is not a practical solution at the university because there are not enough faculty resources to support the satellite mission itself. The improved review process through concurrent documentation work increased the team's understanding of the whole mission simultaneously fulfilling needs for engineering and research, and the review team gained a better understanding of the work.

RQ-3: There was a lack of communication in the decentralized team, limited to weekly meeting or other arranged interactions. Changing to a common working space increased informal communication, and the team is more cohesive. The N2 diagram facilitated communicating understanding of dependencies to the team that was unfamiliar with interdisciplinary projects and described the areas of communication. It is not clear if the introduction of SE tools would have been as successful if the team had not relocated to a common working space. The systemigram has uncovered many of the higher-level modes of interaction in this type of project.

The findings reported in this study has shown the importance of team cohesiveness towards common goals, concurrent with the recommendations from [4]-[9]. Moving into a common working space and having common workshops and informal discussions has facilitated better teamwork and design decisions. The work was naturally scheduled to accommodate exam periods, but the team has not yet experienced the challenge of a long summer holiday break. The process of knowledge transfer from this year's team to next year's will be interesting to study and most likely a challenge. Management of stakeholder expectations has not been addressed in this paper and will be a part of future work. Human interface control was facilitated through the N2 dependency map, and the physical interface control will be further addressed in future work by e.g. Model-Based Systems Engineering (MBSE).

A secondary mission has been introduced that has mission objectives directly derived from some of the PhDs' research objectives. Future work will compare the approach presented in Fig. 3 with directly including mission objectives from the start, to understand how they compare in achieving the research goals and engineering goals concurrently. The latter approach is common for CubeSats and intuitively seems more motivational. Future work will include a continuation of the tools and methods that have been introduced so far, such as the improved research process, A3 walls, collocation of team to facilitate concurrent engineering and N2 mapping. Furthermore, there will be introduction to additional SE tools, especially *requirement management*, *configuration management*, and *MBSE*. MBSE is interesting because it may reduce some of the dependency issues when developing and changing design if the whole team can work on shared model with common attributes. It is expected that the work from Space Systems Working Group (SSWG) [21] will be utilized, as this working group targets academia and CubeSats especially.

Acknowledgements

This work is supported by the Norwegian Research Council (Grant No. 270959), the Norwegian Space Center, and the Centre of Autonomous Marine Operations and Systems (NTNU AMOS). It has been approved by the Norwegian Centre for Research Data. The HYPSO project team is acknowledged for their willingness to participate in this research. The author wants to thank Cecilia Haskins at NTNU for support in developing the systemigrams.

References

[1] Fet, A.M. (1997) "Systems engineering methods and environmental life cycle performance within ship industry." *PhD Thesis.* Trondheim, Norway.

[2] Sopha, B.M., Fet, A.M., Keitsch, M.M., and Haskins, C. (2010) "Using systems engineering to create a framework for evaluating industrial symbiosis options." *Systems Engineering* **13 (2):** 149-160.

[3] Kala, E. "Nanosatellite & cubesat database | missions, constellations, companies, technologies and more." Retrieved from https://www.nanosats.eu/. (Accessed on 11/08/2018).

[4] Grande, J., Birkeland, R., Gjersvik A., Mathiesen, S.V., and Stausland, C. (2017) "Norwegian student satellite program lessons learned." *68th International Astronautical Congress (IAC).* Adelaide, Australia.

[5] Mengu, C., and Masui, H. (2016) "Best practices for successful lean satellite projects." *4th UNISEC Global Meeting*. Varna, Bulgaria.

[6] Alminde, L., Bisgaard, M., Nielsen, J.D., and Bhanderi, D.D.V. (2005) "Experience and methodology gained from 4 years of student satellite projects." *Proceedings of 2nd International Conference on Recent Advances in Space Technologies* **1**: 94-99.

[7] Bouwmeester, J., Aalbers, G.T., and Ubbels, W.J. (2008) "Preliminary Mission Results and Project Evaluation of the Delfi-C3 Nano-satellite." *4S Symposium.* Rhodes, Greece.

[8] Abadi, C.D. "Product and process: an innovative approach for a student satellite program from the systems engineering point of view." *15th Annual/USU Conference on Small Satellites*.

[9] Birkeland, R., and Gutteberg, O. (2013) "Educational aspects of a cubesat project." *2nd IAA Conference on University Satellite Missions and CubeSat Workshop*. Rome, Italy.

[10] Rouse, W.B, Pennock, M.J., and Cardoso, J. (2016) "Advances in sociotechnical systems," in J. Hsu and R. Curran (eds)n *Advances in Systems Engineering*, Reston, VA, USA.

[11] Bromley, D.B. (1990) "Academic contributions to psychological counselling. 1. a philosophy of science for the study of individual cases." *Couns Psychol Q*, **3 (3)**: 299-307.

[12] Yin, R.K. (2018) "Case Study Research and Applications: Design and methods" SAGE, Los Angeles, CA, USA, 6th ed.

[13] Mills, A.J., Durepos, G., and Wiebe, E. (2010) "Encyclopedia of case study research (Vols. 1-0)", SAGE, Thousand Oaks, CA, USA.

[14] Caillaud, E., Rose, B., and Goepp, V. (2016) "Research methodology for systems engineering: some recommendations." *IFAC Conference on Manufacturing Modelling, Management and Control MIM*. Troyes, France.

[15] Haskins, C. (2008) "Systems engineering analyzed, synthesized, and applied to sustainable industrial park development." *PhD thesis*. Trondheim, Norway.

[16] Clegg, B.T., and Boardman, J.T. (1997) "Systemic analysis of concurrent engineering practice" *Factory 2000-The Technology Exploitation Process, Fifth International Conference on (Conf. Publ. No. 435)*, 464-471.

[17] Boardman, J.T. (1995) "Wholes and parts-a systems approach" *IEEE Transactions on Systems, Man, and Cybernetics* **25:** 1150-1161.

[18] Mussante, J., Frittman, J., and Edson, R. "The use of systemigrams in system template development: An example in disaster management". Retrieved from http://anser.org/docs/asyst-doc/Systemigrams-and-Disaster-Management.pdf. (Accessed on 01/31/2019).

[19] Findlay, R., Braukhane, A., Schubert, D., Pedersen, J.F., Muller, H., and Emann, O. (2011) "Implementation of concurrent engineering to phase b space system design" *CEAS Space Journal* **2**.

[20] Viken, A., and Muller, G. (2018) "Creating and applying a3 architecture overviews: A case study in software development." *INCOSE International Symposium.*

[21] "INCOSE – Space Systems Working Group." Retrieved from https://www.incose.org/incose-member-resources/working-groups/Application/space-systems. (Accessed on 11/08/2018).

Paper B

Honoré-Livermore, Evelyn; Haskins, Cecilia, “Model-Based Systems Engineering for CubeSat FMECA,” In: Madni, A.M., Boehm, B., Erwin, D., Moghaddam, M., Sievers, M., Wheaton, M., (eds) Recent Trends and Advances in Model Based Systems Engineering. Conference on Systems Engineering Research, Springer, Cham. [13].

In this paper, we describe an integrated approach to reliability and systems engineering using MBSE.

CSER 2020

Model-Based Systems Engineering for CubeSat FMECA

Evelyn Honoré-Livermore[a]*, Cecilia Haskins[a]

[a]Norwegian University of Science and Technology, 7491 Trondheim, Norway

Abstract

The CubeSat standard has given universities, small companies, developing countries and others a new gateway to space exploration and space knowledge. Combined with shorter development time and Commercial-Off-The-Shelf components the cost has been lowered considerably. However, the combination of use of low-maturity components and inexperienced development teams results in a short lifetime and poor reliability for most CubeSats. The growth of Model-Based Systems Engineering (MBSE) supports re-use of design architectures in many industries and has lowered the costs of development and is gaining popularity in CubeSat teams. This paper demonstrates the application of reliability methods for implementing dependability analysis in MBSE and shows how this can benefit CubeSat teams struggling with limited personnel resources and low experience with space systems.
Keywords: CubeSat, MBSE, model, FMEA, FMECA, risk assessment, systems engineering

1. Introduction

An increasing number of small satellite projects are conducted at universities, many of which do not have previous space hardware or software experience, or relevant curriculum to support the development. There are several reasons for this, such as better access to Commercial-Off-The-Shelf (COTS) components, cheaper launch opportunities, introductory courses from the National Aeronautics and Space Administration (NASA) and the European Space Agency (ESA), multiple papers and guidelines on how to develop and build small satellites, and an increasing demand from students for projects with hands-on experience (Langer et al. 2015; Langer and Bouwmeester 2016; Holtstiege and Bridges 2018; Luther 2016; Berthoud and Schenk 2016; Larsen and Nielsen 2011). Approximately 500 CubeSats are launched each year, where 40% of them are launched by universities (Kulu 2019). However, research shows that the lifetime of a CubeSat mission is short, with over 50% of the satellites DOA (Dead-on-Arrival) (Swartout 2019b).

Efforts to increase the reliability of CubeSats should increase the success rate of missions. Having a more systematic approach to reliability and verification and validation (V&V) of the satellite in early phases is strongly recommended by the literature, as well as learning from other CubeSat or small satellite projects through lessons learned databases or reuse of design. Universities without space experience may not be aware of these resources when

* Corresponding author. Tel.: +47 400 18 389.
E-mail address: evelyn.livermore@ntnu.no

they start a new project and therefore work in a more ad-hoc manner without the rigor and discipline space projects require (Swartout 2019a).

The use of Model-Based Systems Engineering (MBSE) has been promoted as an option to provide a development platform to perform different types of analyses to support the design of a small satellite. The International Council on Systems Engineering (INCOSE) Space Systems Working Group (SSWG) is developing a CubeSat Reference Model (CRM) for this purpose (Kaslow and Madni 2017; Kaslow et al. 2018). By using the CRM as a starting point of a CubeSat project, it is possible to take a more systematic approach to reliability and V&V, as the reference model also includes guidelines on how to develop the CubeSat mission itself and pointers on how to work with V&V.

This paper describes the use of MBSE within a RAM-SE framework for improved reliability analyses through the application of Failure Mode, Effects, and Criticality Analysis (FMECA) and subsequent risk management. The case study is the HYPer-spectral Smallsat for ocean Observation (HYPSO) satellite developed at the Norwegian University of Science and Technology (NTNU) (Honoré-Livermore 2019). The paper builds on the work submitted by three bachelor students in 2019 (Moen, Sjoevold, and Jordheim 2019).

2. Background

CubeSats are satellites built on the standard of a 10cmx10cmx10cm cube. While both NASA and ESA recommend following strict reliability and quality approaches for their long-term missions, these practices often are costly and not realistic for universities. Both organizations have recommendations for CubeSat builders, such as: derating for electrical components; basic Failure Detection, Isolation and Recovery (FDIR) analysis guidelines, applying FMECA to both the functional and physical product tree, and planning a high degree of early testing and verification of subsystems as well as end-to-end functional testing (TEB 2016; CalPoly Sat Program 2017; Capogna and Gupta 2018).

While traditional space projects conducted by NASA/ESA have access to a large knowledge base, experienced people, and may use 5-15 years in development, CubeSat projects approach dependability analysis pragmatically. Access to resources determine the level of analysis performed under conditions of limited knowledge, short schedules (2-3 years), and constrained budgets (Langer and Bouwmeester 2016; Faure, Tanaka, and Cho 2017). In addition, it is typical for CubeSats to use COTS components that have lower reliability, lower Technology Readiness Level (TRL), and less data available to perform the reliability analysis. Performing FMECA on CubeSats has been discussed (Menchinelli et al. 2018) where a practical approach to managing risk through FMECA is described.

CubeSat university projects face several challenges, most notably in access to knowledge and experienced project management, which often directly influence the success of any mission (Honoré-Livermore 2019; Cho 2016; Bouwmeester, Aalbers, and Ubbels 2008). University teams struggle with not having enough people, high turn-over due to graduation, lack of previous experience with project work and multidisciplinary project teams, and finally, but not the least, the challenge of building a satellite to work in space including the stringent requirements for documentation and evidence of space systems testing results.

NTNU has been working on small satellites and CubeSats for the past decades (Grande et al. 2017), but only recently with a mission based on oceanographic research with funding from the Research Council of Norway for both hardware and launch (Honoré-Livermore 2019). Previously, much of the work was conducted in a student organization or loosely associated with course assignments. Today, there is a team of 20-30 undergraduate and graduate students working on developing the HYPSO satellite. As a part of this activity, there is research in systems engineering and project management on how to improve the processes and methods applied in a university-based CubeSat project to increase the probability of mission success and the satisfaction of students and supervisors for their academic work.

3. MBSE for CubeSat

Systems Engineering (SE) emerged from the engineering of complex systems and is today an integral part of product development in many industries and fields of research. Applying Model-Based Systems Engineering (MBSE) as a part of the process is gaining popularity as the tools and methods mature and offer greater interoperability between disciplines in complex product development and lifecycle management. The MBSE approach proposes a unified model that can provide the different viewpoints necessary to perform analyses in all engineering domains (Piggott, Melanson, and Hartman 2007). The viewpoints offer a variety of relevant information and hide the non-relevant data to focus analysis depending on the need (Friedenthal 2016; Haberfellner et al. 2019). Recent advances in the realm of MBSE show increased interoperability between the system model, mechanical analysis tools, electrical analysis tools and other design tools (Brower et al. 2019; Madni and Sievers 2018).

Not all SE tools support the modeling elements needed for reliability analysis but most offer customization options. There are also other approaches to add the failure analysis elements to the model, such as using export/import functions between the system model and the safety model (Sango 2018). A failure mode is the absence of a function or a non-delivery of a need (Schindel 2010), and an added benefit of using MBSE is that the failure analysis can be performed at all phases of the development, as it is possible to identify failure modes at the top level of a system even before the system has been designed. Integrating failure analysis in MBSE can lower costs and time spent on dependability analysis through providing a more systematic framework and increase communication on safety analysis and design. Baklouti et al. discuss the state-of-the-art research of FMECA and MBSE in their paper (Baklouti et al. 2019) and describe a use-case for analysis of an Electro-Mechanical Actuator system. The study by Gregory et al. (2020) recommends the use of MBSE in Functional Avionics in spacecraft for Communication & Consistency and for Template Model Framework. NASA discusses the requirements for a framework for safety analysis of space missions in Evans et al. (2018). This is aligned with ESA and NASA's overall strategy to manage complex space missions with MBSE.

4. RAM and SE framework

Reliability, availability and maintainability (RAM) analysis aims at using engineering knowledge and techniques to control the risk of experiencing failures and to reduce engineering uncertainties (O'Connor & Kleyner, 2012). The main activities of RAM engineering cover (1) artificial experiments to test out the properties of a given system or parts, and (2) analysis and modelling techniques to reveal the cause-effect relationships between failure and specific conditions (Verma et al., 2015).

RAM analysis can be both qualitative and quantitative. Qualitative analysis is used to identify failure modes, mechanisms and causes (such as FMECA), and determine the possible maintenance and test strategies. As the design matures, these analyses may be iterated, and updated via communication and consultation with operators, manufacturers and designers.

Space system design is a concurrent and collaborative process, where different engineering teams are involved. The RAM issues must be considered as early as possible to support the decision making about redundancy, modularization, strategies for interventions and the like. However, the effect of RAM considerations is not easily observed by the whole engineering team, and RAM methods do not have a well-defined interface with other analyses carried out in parallel phases of the design. A similar problem is also identified by Barnard (2008) who points out that the overemphasis on probabilistic modelling frequently leads to misinterpretation of RAM analysis, which can lead to bad design or waste of engineering efforts. A recently proposed RAM-SE framework recommends several activities to integrate both the SE and RAM community as shown in Fig. 1(a) (Zhang et al. 2018).

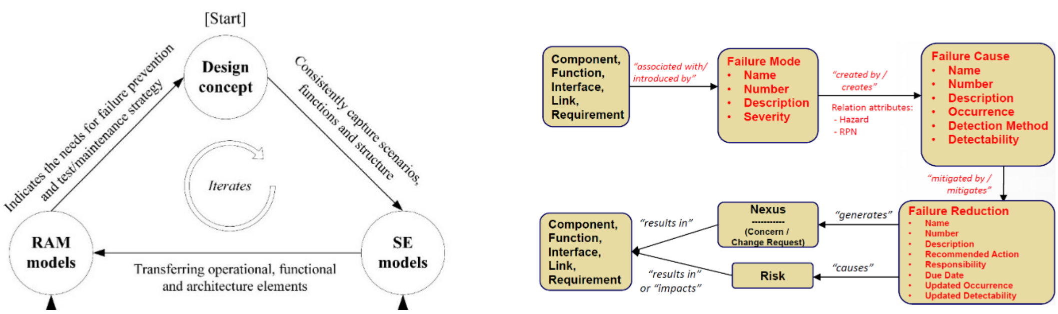


Fig. 1 (a). Conceptual RAM-SE (Zhang et al. 2018). (b) Classes used for modelling from (Kratzke 2018). The classes in black are the existing ones in the tool, while the classes in red are the ones that needed to be added. Some of the attributes given in red already existed, while others had to be added

Estimation of the reliability of the components and in-house developed software is done by the team when better sources of data are not available. The results of the analysis are thus dependent on the expertise and configuration of the team and may not be valid. However, the exercise of performing the analysis and the ranking of critical failure modes and causes are valuable because they create awareness in the team of potential issues in the design. In addition, the cognitive process of thinking about dependability and reliability can lead to better design processes and decisions in the future.

An FMECA analysis can be categorized broadly into two different types based on the approach: top-down functional FMECA, or bottom-up component FMECA – given from MIL-STD-1629A (Department of Defense 1980). Depending on the phase of the project and the maturity of design and other information available, one is more appropriate than the other. Using a model-based approach allows for a unification of these types.

When the system is continuously modeled from the operational, functional, logical and physical viewpoint, it is possible to combine the different types of analysis and aggregate the data. The operational viewpoint is implemented through the functional design which in turn is allocated onto logical units that are finally realized in the physical design. Depending on the tools used for modeling, real data from testing and verification can also be introduced and give a fifth viewpoint and source of information for the dependability analysis (Schindel 2010; Bürger 2019). Table 1 indicates benefits of integrating RAM and SE models.

Table 1 Advancements for RAM methods in SE context (Zhang et al. 2018)

Methods	Objectives	Advancements of systems thinking
FMECA	- Uses a basis for detailed RAM analysis and maintenance optimization and planning - Document the effect of failure on system	- Systematically identify all operational modes and functions attached to each potential failure modes - Carry out an extended/revised type of FMECA that is able to involve dynamic aspects of key scenarios, see also the discussion in (Issad et al., 2017)

Incorporating RAMS aspects as early as possible gives several advantages in form of engineering efforts and budgets in many industry sectors such as nuclear, satellite and aviation, where the analysis is further amplified by the complexity of design solutions. (Zhang et al. 2018)

5. Modeling and FMECA implementation

The SE tool used to implement FMECA in this research did not have the necessary modeling elements natively. The classes that had to be added to the modeling tool were based on a presentation given by Kratzke (2018) shown in Fig. 1(b). These classes were added to represent failure mode, failure cause, failure reduction, and the relationships between these and with the existing classes. The modeling and FMECA was performed on the NTNU HYPSO mission. The CubeSat consists of multiple subsystems, shown in Fig. 2. The payload (PLD) subsystem interacts with the subject of interest and is the most critical part of the space segment. The payload consists of several subsystems: HyperSpectral Imager (HSI), RGB camera (RGB), On-board Processing Unit (OPU) and Break-Out Board (BOB).

System to be analyzed: HYPSO CubeSat mission. The mission of the satellite is defined by a mission statement and mission success criteria. These are the most important requirements of the mission, and therefore, the FMECA analysis chose to focus on these. The system was modelled in a MBSE tool, with requirements, component trees, functional trees and chains, and operational user scenarios. Since the HYPSO team is tasked with developing the payload systems and ground segment while the other subsystems are COTS components with a higher TRL, the analysis was limited to the space segment payload systems and the ground segment.

FMECA workshops: FMECA workshops were conducted during which the HYPSO team members were asked to identify the operational modes for each of the subsystems, and which functions were necessary for the operational modes that would then be used as a basis to identify the failure modes. Using the mission success criteria, which are reflected in the concept of operations of the system, the FMECA focused on the critical parts of the performance of the system.

Failure mode assessment: The assessment followed a procedure tailored from the aforementioned Mil-Std-1629A (Moen et al. 2019). Each failure mode is assessed with respect to severity (scale 1-5, where 1 is negligible and 5 is absence of function)), and occurrence (scale 1-5, where 1 hardly ever occurs, and 5 is probable). This analysis provides the criticality number, given by the multiplication of severity and occurrence. Next, the failure modes are assessed according to their detectability, (scale 1-10). A high detectability is given index 1, meaning that it is easy to identify the failure – i.e. that only one type of failure can give the effect that is observed. For many CubeSats, having a high level of detectability is more important than avoiding all failure modes, especially if there is a method to reset or remove the failure. Identifying the failure mode can also allow for redesign of the system in the next satellite. The attributes are evaluated and added to the model and used to calculate the Risk Priority Number (RPN), which is given by the multiplication of severity, occurrence and detectability.

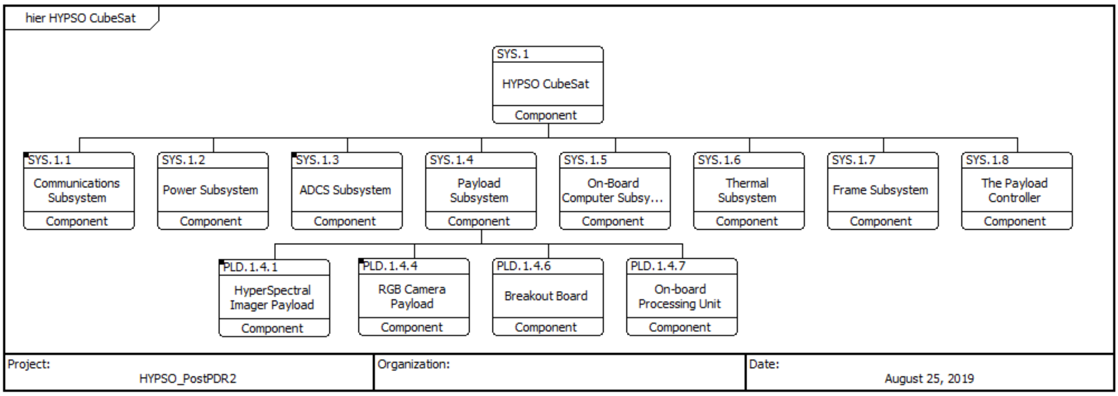


Fig. 2. The HYPSO CubeSat physical hierarchial structure in MBSE.

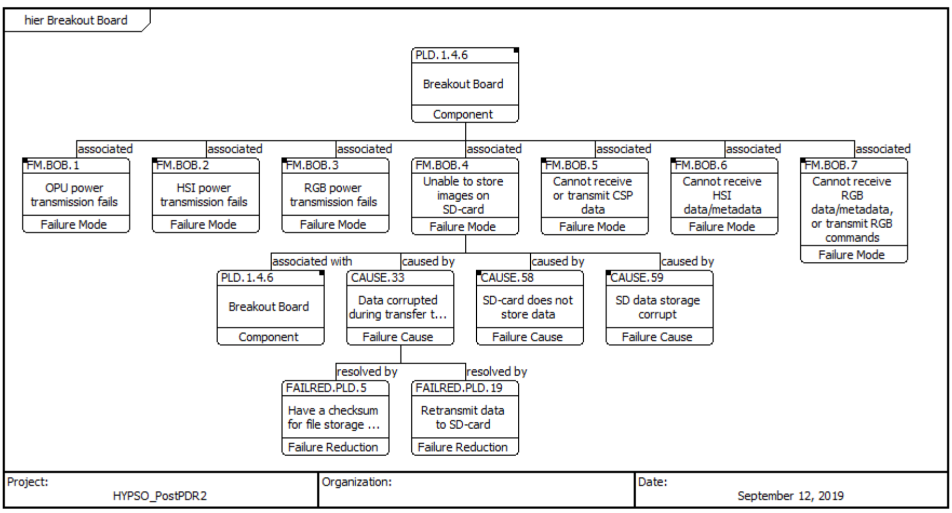
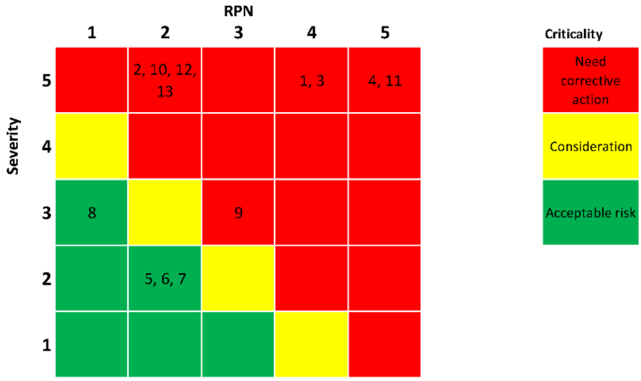


Fig. 3. BOB with its failure modes displayed in a hierarchy diagram. One of the failure modes has been expanded to show the failure causes and reduction methods.



ID	Risk Reference
1	FM.BOB.6-CAUSE.40
2	FM.BOB.6-CAUSE.39
3	FM.BOB.5-CAUSE.31
4	FM.BOB.5-CAUSE.4
5	FM.BOB.4-CAUSE.58
6	FM.BOB.4-CAUSE.59
7	FM.BOB.4-CAUSE.33
8	FM.BOB.3-CAUSE.30
9	FM.BOB.3-CAUSE.27
10	FM.BOB.2-CAUSE.36
11	FM.BOB.1-CAUSE.27
12	FM.BOB.1-CAUSE.30
13	FM.BOB.2-CAUSE.30

Fig. 4. Risk matrix for BOB failure modes. The RPN is referenced to a scale of 100, and then into ranges of 20 to give the resulting 1-5 indexing .

A total of 69 failure modes, 65 causes and 45 failure reduction actions were identified and modelled. The PLD subsystem BOB is used here as an example of the actual implementation, see Fig. 3 and Fig. 4. The severity, occurrence and detectability attributes were determined based on the workshop input, resulting in ranked failure modes. These attributes are assumed to be evaluated for nominal operations, not for ground testing or integration.

The FMECA model and attributes are used to generate a risk matrix composed of RPN and severity. The risk matrix includes severity as a high RPN number may not indicate a true risk to the project. The risk matrices are given per subsystem to make it simpler for the team to use. The critical failure causes and their corresponding reduction methods should then be prioritized in project work.

6. Discussion

The process identified 13 risks for BOB. Of these, 8 were of severity rating 5. The risk matrix in Fig.4 was configured such that all causes with severity rating 5 would need corrective action, as this indicates a non-fulfillment of minimum mission success criteria. Prior to the workshop and modeling, only the loss of power transmission had been considered as a risk in design discussions. Based on this analysis, the BOB component has been updated to reduce the number of critical risks through implementation of failure reduction measures. This process agrees with the RAM-SE process proposed in Fig. 1(a) where the design concept is modeled in MBSE and the subsequent RAM analysis is integrated in the model.

The modeling tool supports the inclusion of actors (humans) and their influence on the system through e.g. use cases, but this was not considered in the modeling. Human error can be a considerable cause of failure modes to the system. The focus on mission success criteria when analyzing the system also left out an important part of the CubeSat development lifecycle: test and verification. Use cases can be developed for test activities such as shock or vibration, which carry high risk of damage to the device-under-test (DUT) and personnel. Including these use cases for future CubeSat modeling would address additional risk areas. Additionally, the choice of perspective when modeling and analyzing did not support evaluation of simultaneous failure. A Fault Tree Analysis (FTA) on a functional level would increase the chance of discovering if two faults would give a severe impact that they individually would not in the FMECA. This is supported in the class relationship implemented according to Fig. 1(b). Limited knowledge about the reliability of low-TRL components lowers the validity of the analysis. It is largely dependent on the people attending the workshop and their ability to properly assess the severity, occurrence and detectability of individual events. Modeling facilitates visualization of the relationships between the failure modes, the components, the functions and the overall success criteria.

For CubeSat teams the starting point of a reference model such as the CRM provided by INCOSE SSWG lessens the burden on the faculty to have prior experience with satellite systems and focuses the team work on specific subsystems. The university teams can then populate the model with the known information such as regulations, existing ground systems, communication capabilities and operational aspects specific to the organization. Re-use of a model with failure analysis built-in from the previous mission or other teams' missions will reduce the team resources needed to perform dependability analysis. An added benefit from using MBSE is the potential of automated analyses on the same model. Each element in a model will have attributes that can be analyzed automatically through scripts or functions in the tool. For dependability analysis, it is then a matter of checking what the change of one element will have on the overall dependability of a system.

Swartout (2019a) highlights the need for "streamlined practices, experientially developed" and recommends increasing the time spent on integration and test, also supported in (Faure et al. 2017). Designing the system in MBSE through the application of a CRM and incorporated dependability analysis could be a part of the "streamlined practices" to improve the design maturity faster, thereby allowing more time to perform integration and test.

7. Conclusion

Developing and building complex systems is a challenge, and while CubeSats are small, they are still complex systems requiring a systematic engineering approach. Statistics show that there are still many CubeSat missions that fail, mostly because of low dependability. However, performing the full dependability and quality management that NASA and ESA recommend is not feasible for small university teams lacking both people, money, and time. A pragmatic approach to dependability analysis through the usage of MBSE systems that can support the FMECA analysis offers promise to increase dependability for university CubeSat teams.

This paper has shown how the framework suggested by Kratzke (2018) can be applied to an existing CubeSat MBSE model, and support the subsequent risk management from the FMECA workshops. The FMECA workshop and modeling enabled the HYPSO team to understand which failure modes are associated with different subsystems and how these affect the overall success criteria for the mission. It also established a priority for completing different failure reduction activities and highlighting critical interfaces and components/functions. Using visualization through the operational scenarios modeling and focusing the effort on making the mission a success has proven to be of great value to the CubeSat team that did not have previous experience with dependability analysis. It also enables keeping the information connected and ensures traceability for future design decisions.

The results agree with the suggestions of Gregory et al. (2020); MBSE has been used as a tool to improve Communication and Consistency of the RAM (project) information, and it can be re-used in future spacecraft building on the same spacecraft subsystem structure with little rework necessary. The framework could be a part of the university MBSE template or used to extend the CRM.

Incorporating the CubeSat Reference Model with FMECA analysis from an operational or functional top-down view, or component-based bottom-up view has required less effort than starting from scratch. Furthermore, the analysis could be performed at different phases of the development, as the failure modes are relevant on higher operational levels as well as lower functional or logical levels. Re-use and continuous development of the systems is facilitated using MBSE as suggested by Madni and Sievers (2018).

For this preliminary work, the risk assessment was done in a separate tool from the MBSE tool because the research team uses a project management suite. As future work, this could be automated and linked to improve workflow. Additionally, future work should explore how the lessons learned and failure analysis from multiple teams could be aggregated into the reference model, highlighting typical failure modes that are associated with certain interfaces or types of subsystems.

Acknowledgements

This work is supported by the Norwegian Research Council (Grant No. 270959), the Norwegian Space Agency, and the Centre of Autonomous Marine Operations and Systems (NTNU AMOS).

References

Baklouti, A., Nguyen, N., Mhenni, F., Choley, J., Mlika, A., 2019. "Improved Safety Analysis Integration in a Systems Engineering Approach", Applied Sciences, 9.

Barnard, R. W. A. (2008). What Is Wrong with Reliability Engineering? Paper presented at the Proceedings of the 18th Annual INCOSE International Symposium, Utrecht, the Netherlands.

Berthoud, L, Schenk, M., 2016. “How to Set Up a CubeSat Project – Preliminary Survey Results.” In 30th Annual AIAA/USU Conference on Small Satellites. Logan, UT.

Bouwmeester, J., Aalbers, G.T., Ubbels, W.J., 2008. “Preliminary Mission Results and Project Evaluation of the Delfi-C3 Nano-Satellite.” In 4S Symposium Small Satellites Systems and Services. Rhodes, Greece

Brower, E. W., Delp, C., Karban, R., Piette, M., Gomes, I., Wyk, E., 2019. “OpenCAE Case Study: Europa Lander Concept.” In INCOSE International Workshop. Torrance, CA.

Bürger, E.E., 2019. “A conceptual MBSE framework for satellite AIT planning”. Doctoral thesis. Instituto Nacional de Pesquisas Espaciais.

California Polytechnic’s PolySat Program, 2017. CubeSat 101. California Polytechnic State University at San Luis Obispo.

Capogna, F., Gupta, V., 2018. “How to increase CubeSat reliability!” ESA (TEC-QCD).

Cho, M., Mazui, H., 2016. “Best practices for successful lean satellite projects.” In 4th UNISEC Global Meeting.

Defense, Department of. 1980. “MIL-STD-1629A Procedures for performing a Failure Mode, Effects and Criticality Analysis.” Washington, DC.

Evans, J., Groen, F. J., Wang, L., Okon, S., Austin, R., Witulski, A., Mahadevan, N., Cornford, S., Feather, M., Lindsey, N., 2018. “Towards a framework for reliability and safety analysis of complex space missions”, International Journal of Human Factors Modelling and Simulation, 6: 203.

Faure, P., Tanaka, A., Cho, M., 2017. “Toward lean satellites reliability improvement using HORYU-IV project as case study”, Acta Astronautica, 133: 33-49.

Department of The Air Force. 1988. “MIL-STD-1543B Reliability program requirements for Space and Launch Vehicles.” Washington, DC.

Grande, J., Birkeland, R., Gjersvik, A., Mathisen, S., Stausland, C., 2017. “Norwegian student satellite program – lessons learned.” In 68th IAC.

Gregory, J., Berthoud, L., Tryfonas, T., Rossignol, A., Faure, L., 2020 “The long and winding road: MBSE adoption for functional avionics of spacecraft”, Journal of Systems and Software, 160, art. no. 110453

Haberfellner, R., de Weck, O., Fricke, E., Vössner, S., 2019. “The Systems Engineering Basics in Our Systems Engineering Concept.” in, Systems Engineering: Fundamentals and Applications (Springer International Publishing: Cham).

Holtstiege, J., Bridges, C., 2018. Lean satellite design for amateur communications payload in the ESA ESEO mission.

Honoré-Livermore, E., 2019. “CubeSats in University: Using Systems Engineering Tools to Improve Reviews and Knowledge Management”, Procedia Computer Science, 153: 63-70.

Issad, M., Kloul, L., & Rauzy, A. (2017). A scenario-based FMEA method and its evaluation in a railway context. Paper presented at the Reliability and Maintainability Symposium (RAMS).

Kaslow, D., Ayres, B., Cahill, P., Hart, L., Levi, A., Croney, C., 2018. “Developing an MBSE CubeSat Reference Model – Interim Status #4.” In 2018 AIAA SPACE and Astronautics Forum and Exposition. Orlando, FL.

Kaslow, D., Madni, A., 2017. “Validation and Verification of MBSE-compliant CubeSat Reference Model.” In Conference on Systems Engineering Research, edited by Barry Boehm Azad M. Madni. Redondo Beach, CA, USA.

Kratzke, R., 2018. “Failure Modes Effects Analysis in MBSE.” In INCOSE Texas Gulf Coast Chapter. Houston, TX.

Kulu, E., 2019. “NanoSats database”, Accessed June 2019. https://www.nanosats.eu/.

Langer, M., Olthoff, C., Harder, J., Fuchs, C., Dziura, M., Hoehn, A., Walter, U., 2015. “Results and lessons learned from the CubeSat mission First-MOVE.” In Small Satellite Missions for Earth Observation, 10th International Symposium, IAA. Berlin.

Langer, M., Bouwmeester, J.. 2016. “Reliability of CubeSats – Statistical Data, Developers' Beliefs and the Way Forward.” In AIAA/USU Conference on Small Satellites.

Larsen, J., Nielsen, J.D., 2011. “Development of cubesats in an educational context.” In International Conference on Recent Advances in Space Technologies, 777-782. Istanbal, Turkey: IEEE.

Luther, S. T., 2016. “SysML Based CubeSat Model Design and Integration with the Horizon Simulation Framwork”, Master of Science, California Polytechnic State University.

Madni, A., Sievers, M., 2018. “Model-Based Systems Engineering: Motivation, Current Status, and Needed Advances.” In Disciplinary Convergence in Systems Engineering Research. Springer Intenrational Publishing.

Menchinelli, A., Ingiosi, F., Pamphili, L., Marzioli, P., Patriarca, R., Costantino, F., Piergentili, F., 2018. “A Reliability Engineering Approach for Managing Risks in CubeSats”, Aerospace, 5: 121.

Moen, A. K. A, Sjøvold, E., Jordheim, O.. 2019. “Implementation of FMECA in Small Satellite Development”, Norwegian University of Science and Technology. Bachelor thesis.

O'Connor, P., & Kleyner, A. (2012). Practical Reliability Engineering, 5th Edition. Hoboken, NJ: John Wiley & Sons, Inc.

Piggott, S, Melanson, P., Hartman, L.. 2007. “A Vision for Super-Model Driven Systems Engineering'”, INCOSE International Symposium, 17: 52-65. doi:10.1002/j.2334-5837.2007.tb02857.x

Requirements and Standards Division, ESA-ESTEC. 2009. “Space product assurance: Dependability, FME(C)A.” Noordwijk, NL.

Friedenthal, S., Oster. C., 2016. “Applying SysML and a Model-Based Systems Engineering Approach to a Small Satellite Design.” in Richard Curran John Hsu (ed.), Advances in Systems Engineering (American Institute of Aeronautics and Astronautics: Reston, Virginia, USA).

Sango, M., 2018. “Model Based Safety Analysis Thanks to the Bridge Between Capella and Safety Architect.” Accessed August 2019, https://www.youtube.com/watch?v=-NtUKdaUGdc. YouTube.

Schindel, W. D., 2010. “Failure Risk Analysis: Insights from MBSE.” INCOSE International Symposium, 20: 1227-1239.

SEMATECH. 1992. “FMEA: A Guide for Continuous Improvement for the Semiconductor Equipment Industry.”

Swartout, M., 2019a. “CubeSat Mission Success: Are We Getting Better?” In 2019 CubeSat Developers’ Workshop. San Luis Opisbo, CA.

Swartout, M. 2019b. “Mission Success for Universities and Professional Programs”, Accessed September 2019. https://sites.google.com/a/slu.edu/swartwout/home/cubesat-database/repeat-success.

TEB. 2016. “Product and Quality Assurance Requirements for IOD CubeSat Projects. TEC-SY/129/2013/SPD/RW.” Noordwijk, NL.

Verma, A. K., Ajit, S., & Karanki, D. R. (2015). Reliability and Safety Engineering. United Kingdom: Springer London Ltd.

Zhang, J., Liu, Y., Lundteigen, M.A., Haskins, C. (2018) The Application of Systems Engineering for Framing Reliability, Availability and Maintainability in Subsea Design. Systems Engineering. DOI: 10.1002/sys.21462

Paper C

Honoré-Livermore, Evelyn; Bakken, Sivert; Prentice, Elizabeth Frances, "Factors Influencing the Development Time from TRL4 to TRL8 for CubeSat Subsystems at a University," 9th International Systems & Concurrent Engineering for Space Applications Conference (2020) (virtual) [14].

In this paper, we describe factors that influence the development time of CubeSat subsystems at a university. The findings are based on a series of interviews and experiences from the team members.

FACTORS INFLUENCING THE DEVELOPMENT TIME FROM TRL4 TO TRL8 FOR CUBESAT SUBSYSTEMS AT A UNIVERSITY

Evelyn Honoré-Livermore[1], Elizabeth F. Prentice[2], and Sivert Bakken[2]

[1]*Department of Electronic Systems, Norwegian University of Science and Technology, 7491 Trondheim, Norway*
[2]*Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7491 Trondheim, Norway*

ABSTRACT

It is challenging to estimate the development time of subsystems for CubeSats with low maturity. At the same time, in order to secure funding, and to not run out of funding, estimating the development time is an important part of planning a CubeSat project. We have looked at factors that influence the development time of a hardware-software payload system from TRL4 to TRL8, with the goal of integration into a CubeSat bus for a research mission. Our analysis showed that the most critical factors affecting the development time were clear objectives, internal communication and team knowledge. A tentative relationship between maturity level and factors is presented to help university project managers plan resource needs and mitigating activities.

1. BACKGROUND

CubeSat projects at university have gained tremendous popularity the past decades. The objectives of the projects have been two-fold, providing experience of space systems and project to students, and providing scientific data from CubeSat missions such as Earth Observation. Some of the challenges these projects experience are project management; balancing coursework and satellite work; high turnover; and ensuring mission success [1]. Berthoud et al. (2019) discuss CubeSat project management based on three university case studies. They saw that there was a gap of 2.1 years [2] between industry and university CubeSat development time, and that there is a large number of "lessons learned" papers published on different aspects of university CubeSat projects. Among their findings from the case studies, the authors highlighted several sociotechnical characteristics of the team structure and project management (such as motivated staff, a nominal lifecycle of 2–3 years, passionate students, mixing coursework and non-coursework, version-control), and a strong emphasis on testing and integration to achieve a successful mission.

Aforementioned challenges result in cost or schedule overruns and add to the difficulty to manage and enable estimation of the development time of CubeSat subsystems at university. To achieve better estimates, we first need to understand the factors influencing the development time.

1.1. Product Development Time

To estimate the project schedule, the scope of work, resources available and funding available must be estimated and assessed. For CubeSat projects in academia, the resources commonly include volunteer students, faculty members, engineering support and researchers or Ph.D. students [3, 4]. The amount of funding needed depends on the mission of the project, and can for example be provided through governmental funding bodies or industry support. The scope of work relates to the phases the project encompasses, and the maturity of the systems involved and which maturity level they should reach within the project. It is common to re-assess the schedule regularly to communicate with stakeholders and ensure project success, as sources of funding may run out or resources leave the university or become unavailable. The lifecycle of a system is commonly divided into phases such as (1) Definition and analysis phase; (2) Technology development; (3) Engineering and manufacturing development or detailed design definition; (4) Production and deployment, and (5) Operations and support [5]. For this paper, we are concerned with phases (1)-(4), which commonly correspond to the time before launching and operating the spacecraft.

1.2. Maturity Levels

Maturity levels of technology, integration and system have been used to make more informed estimations of schedule. These are known as *Technology Readiness Level (TRL)*, *Integration Readiness Level (IRL)*, and *System Readiness Level (SRL)* [6, 7]. Originally developed for defense acquisitions, maturity levels have been adopted European Space Agency (ESA) and National Aeronautics and Space Administration (NASA) in their projects and are being used in other commercial and non-commercial sectors as well [8]. TRL is the most widely known and used maturity level, CubeSat subsystem vendors use TRL in product data sheets to provide information to their customers, and most can easily use the level when assessing which subsystems to purchase for their system. Some associate maturity levels with the product development lifecycle [9], which can help management determine if a gateway is completed successfully or not.

Funding bodies and larger defense companies may require maturity level assessments as a part of evaluating the project, and have procedures and methodologies in place for this [7, 9]. Weiping et al. (2011) discuss different approaches to using maturity level assessments and limitations of these. The most apparent limitation is how IRL and TRL are assessed by *subject matter experts* supported by comparison to previous work, established standards and **gut feelings**. Furthermore, the maturity levels themselves use wording that is open to interpretation depending on the person assessing and the context.

The TRL of a technology is assessed on a scale from 1–9, by "subject matter experts" [9, Table I], where the context and environment that the technology shall be used in becomes relevant at levels 5 and up. The IRL is also assessed on a scale from 1–9 by subject matter experts, which ranges from definition of the interface through structured communication to success in a deployed system [9]. The SRL is given by a combination of the TRLs of the technologies involved, and the IRL. Tompkins et al. (2020) list the major four calculation methodologies for determining the SRL, where the Sauser SRL method is given in Eq. 1.

$$SRL = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{m_i}\left[\left(\frac{1}{9}IRL_{SRL}^{n\times n}\right) * \left(\frac{1}{9}TRL_{n,1}\right)\right]\right] \tag{1}$$

where n is the number of elements in the TRL vector, each represented as TRL_i, the IRL_{ij} is measured as an integration between elements i and j. The factor m_i is the number of integrations per element (i.e. the non-zero elements in each row in the IRL matrix, see Eqs. 4 and 5. The IRL_{ii}, i.e. the integration of an element with itself, is assigned a level of 9. The values are then normalized and the SRL matrix is calculated by taking the matrix product of the IRL matrix and the TRL vector. Following the procedure of [9], we show the building blocks of Eq. 1.

$$[SRL]_{n\times 1} = [IRL]_{n\times n} \times [TRL]_{n\times 1} \tag{2}$$

The SRL of an element is given in Eq. 3, a combination of the maturity of the specific element or technology and its interfaces to the adjacent elements.

$$[SRL] = \begin{bmatrix} SRL_1 \\ SRL_2 \\ \dots \\ SRL_n \end{bmatrix}$$

$$= \begin{bmatrix} IRL_{11}TRL_1 + IRL_{12}TRL_2 + \dots + IRL_{1n}TRL_n \\ IRL_{21}TRL_1 + IRL_{22}TRL_2 + \dots + IRL_{2n}TRL_n \\ \dots \\ IRL_{n1}TRL_1 + IRL_{n2}TRL_2 + \dots + IRL_{nn}TRL_n \end{bmatrix} \tag{3}$$

We are interested in the full SRL of our system, which is the arithmetic mean of each of the element SRL. The element SRL can be calculated by normalizing the SRL value, shown in Eq. 4 where m_i is the number of integrations of the element i including its own interface.

$$SRL_{element} = \frac{SRL_i}{m_i} \tag{4}$$

To get the composite SRL, we take Eq. 4 and calculate the average for each of the element SRL, shown in Eq. 5.

$$SRL_{comp} = \frac{\frac{SRL_1}{m_1} + \frac{SRL_2}{m_2} + \dots + \frac{SRL_n}{m_n}}{n} \tag{5}$$

By combining Eqs. 2-5, we get Eq. 1, which is used to assess the maturity of a system at different phases in the lifecycle. The SRL is a numerical value from $0 - 1$, where 1 is full maturity, where we can expect that it is operationally deployed.

2. METHODS

We have followed a university CubeSat team from Phase 0/A to Phase D/E, in which the payload subsystem has been developed from TRL4 to TRL8. The payload includes both software and optomechanical hardware. The university CubeSat team consists of 20 students and 6-8 Ph.D. and Post.Doc. fellows. The students join for 1-2 semesters as part of their thesis.

The payload of the CubeSat under study is a hyperspectral imager with ambitious on-board processing capabilities. With the appropriate processing the mission aims to increase the response times and save bandwidth for operations. Low-level programming and specialized hardware is needed to achieve this with the chosen on-board computer that has limited resources in terms of power and computational capacity. While the concept of splitting light into a spectrum of wavelengths is not new (TRL1), the chosen instrument and optical design was TRL3/4 at the beginning of the project. The imager is based on Commercial-Off-The-Shelf (COTS) components and worked in a laboratory setting.

2.1. Case Study

The method used for the case study has been action research [10, 11]. The authors are active participants in the project, and have project management and subsystem management roles, which includes continuous improvement of processes and procedures. Data sources include project documentation, statistics from GitHub and semi-structured interviews (n=5) with some of the team members focused on interfaces and product development.

Technology development followed closely from the work of two main teams – software and hardware. Beginning at TRL4, these two teams worked primarily separate and technology grew independently. In order to approach TRL6, coordination between the two teams was necessary in building subsystem and system level prototypes. Two case studies on technology development are presented, one from each team. The studies illustrate that although development grew from separate sources and teams, many of the same factors influenced the pace of the project.

The semi-structured interviews were conducted with representatives which had been a part of integration activities at different periods of the project. The interviewees were asked to describe their subsystem, if they were able to draw a block diagram with its interfaces, describe the different types of interfaces, the information sources used to understand the interfaces, if they had utilized sequence diagrams or Interface Control Document (ICD)s, and what was challenging or easy with the development of their subsystems.

2.2. Assessing the Readiness Levels

We assessed the TRL and IRL of the system at the current status, in lifecycle phase (3), as the project has not reached phases (4)-(5) yet. For the purpose of providing information to estimating development time and schedule, the limitations of subjectivity and bias when assessing maturity levels become less problematic when used for internal evaluation and communication because it will be the same people doing relative assessment of the technologies over a longer period of time [12].

3. RESULTS

The original project schedule was made in late 2017, before the project had its Mission Design Review (MDR) and before deciding which spacecraft bus and subsystems to integrate the payload with. The project schedule has been revised at the major gateways (Preliminary Design Review (PDR) in December 2018, a new PDR in June 2019, and Critical Design Review (CDR) in March 2020), in addition to small adjustments continuously.

This section describes the development of software and hardware, and identifies the main factors influencing the development time. The semi-structured interviews are used to illustrate how a typical team member experiences the development process.

3.1. Software team development

In this CubeSat project, students and staff have contributed to different software stack components i.e. Hardware, Operating Systems, Middleware, Applications, and user interface layer. To achieve this a lot of development considerations needed to be made, agreed upon and followed-up to ensure the desired level of coherence and maintainability of the software stack. Initially, there was no such coherent workflow available to guide the students in the development and the documentation and code were less intelligible as a result. The need for such a workflow became evident after software PDR.

Thus, a workflow was proposed to better guide the software stack development coherently. This is now a part of the on-boarding procedure for new students. Given the limited time students can commit to the project, some only for one semester, the workflow needed to be attainable. That is, there was a limitation within the workflow to utilize development tools that the students should be familiar with concerning programming language, revision control system, build environment, and so on. The first batch of students that were introduced to the baseline workflow found that it took some time to adapt, but that the benefits in terms of collaborative work made it worthwhile. Through the use of the workflow, in close collaboration with other subsystem teams, it has become more familiar and better exploited across the team.

Most incoming students on the team had some experience with the programming languages C/C++ for low-level programming from university classes. Embedded systems, such as the one planned for the CubeSat under study, have a functional compiler in C, and the support for C found in Application Programming Interface (API) is not as common as for other low-level languages. There are some challenges related to using low-level programming and specialized hardware. The C language can be hazardous as it provides the programmer more direct control over the memory usage and run-time behavior. This can again provide better utilization of the computational capacity available. There are *safer* languages than C that rely on a larger runtime, a more complicated feature set, and maybe even virtual machines to work. However, the compilers available for C perform well even with a limited computational capacity. Using a *safer* language would result in a longer onboarding process. Through the use of such tools in the software development stack as compiler warnings, linters[1], and other static analysis tools to detect preventable issues the potential pitfalls of C can to some extent be mitigated.

Another major point was finding a shared platform to develop on. Git by GitHub was the chosen high-level tool used for the development of all software and issue tracking, where the branching strategy known as GitHub Flow was selected [13]. The software CDR highlighted the importance of using GitHub and its functions. This has also been used to enable a scrum approach to software development [14]. The issue tracking provided a common platform to discuss software development, making it easier to talk about software, document software, ask questions, or request new features. The GitHub Flow branching strategy aims to have a working master branch with as little overhead as possible. This lessened some complications with integration as the developers were continuously reminded that their contributions were expected to be deployed on the target hardware.

Specialized hardware that enables the use of field-programmable-gate-arrays, i.e. re-configurable hardware logic, is used to further expand the onboard processing capabilities. This comes at the cost of increased complexity in terms of development and integration. Developing in hardware description languages is time demanding, and does not encourage a lot of online processing flexibility. Development of software tests, and writing

[1] A tool that analyzes code to find errors and bugs.

code that is modularized and testable can and has been shown to further improve the robustness of the deployed software. However, this can be hard to motivate in general, and especially as part of academic work. Prioritizing contributions to the software stack versus progression on personal academic work has proven to be a challenge, but the team has developed strategies to find synergies between the two.

3.2. Hardware team development

Hardware development was divided into two phases, the technology development phase where much of the design work was completed, and the engineering and manufacturing phase where assembly and testing occurred. Through these phases we were able to push our technologies through TRL levels. Hardware development focused on the primary payload, or hyperspecral imager. More details on development follows.

Technology development phase: The TRL4 model was a functioning hyperspectral imager used for desktop measurements or unmanned aerial vehicle flights. However, its optics had been especially designed for imaging from low earth orbit. The lab was an empty room, available facilities on campus were yet undiscovered by the team, no one had training to use other facilities, there was no database nor documented guidance on how to proceed, and nothing had been purchased so there was no relationship with industry suppliers. Very few students involved had prior experience in the CubeSat discipline nor did the project advisors. At this point, a lot of time was spent planning and defining requirements both for the mission and systems on board. At this time, the hardware team focused on a full exploration of the capabilities of the imager along with understanding its assembly/disassembly and nuances.

At TRL5, the hyperspectral imager had been dissected into components and much of the time went into understanding if individual parts could withstand the space environment. Here, individual lenses, detectors, etc. were tested in vacuum and thermal chambers. Design modifications and new assembly procedures were developed based on experience gained from testing.

Engineering and manufacturing phase: Coordination with the software team became extremely important to achieve TRL6. Hardware and software were combined to demonstrate a working prototype. Here, many missing or faulty interfaces, cables, connectors, etc. were discovered. The work at this stage was severely underestimated, even down to the timeline of getting parts machined in-house. In addition to getting the payload working and mounted in the satellite, the prototype also went through basic environmental testing to prepare for design qualification.

To reach TRL7, is the completion of environmental testing on the qualification model. At this stage, most parts had already been ordered, test objectives outlined, test plans written, and the design was frozen. Most time went into waiting for machine time at external facilities and working through test plans and reports.

As with software, the major factors influencing each of these stages in hardware development can be broken down into categories: internal facilities (lab and university), team knowledge, parts supply chain, clear objectives, and external facilities.

The chart in Fig. 1 illustrates a breakdown of the primary factors that influenced each TRL for the hyperspectral imager development from a hardware perspective.

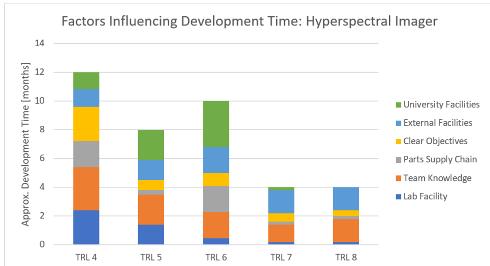


Figure 1: Factors influencing the development time of the hyperspectral imager.

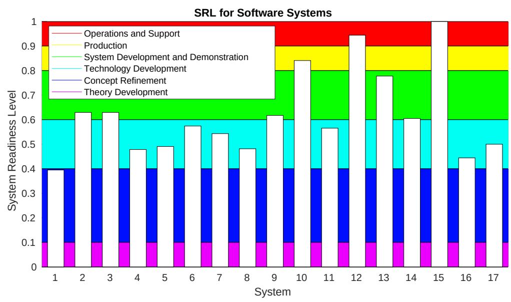


Figure 2: System Readiness Levels for software modules. Systems 1-9 are NTNU payload, systems 10-14 are spacecraft software modules, and systems 15-17 are ground software modules.

In summary, important factors influencing hardware and software development are:

1. **Lab facility:** Target Hardware set up correctly, what tools and machines are in the lab, design and testing software, data storage and management, spare hardware parts and materials, ESD protected areas[2], special corona restrictions and control, remote access to target hardware. Available target hardware for testing.
2. **University facilities:** machine availability, prioritization of projects, training and access required to

[2]Where all surfaces, people and objects are kept at the same potential to avoid damage to electronics.

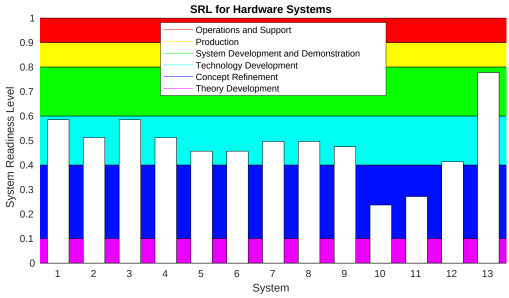


Figure 3: System Readiness Levels for hardware. Systems 1-6 are NTNU payload, systems 10-13 are the thermal, vacuum, radiation and vibration subsystems.

use the facilities, special corona restrictions and control.

3. **Team knowledge:** Prior knowledge from existing team members and advisors, a database of documentation or templates, professional network, and ways to reach them with questions, a common workflow.
4. **Parts supply chain:** shipping procedures, known suppliers, special suppliers for space materials, invoicing, methods for speeding up delivery of parts if necessary.
5. **Clear objectives:** A roadmap of the project, well-defined roles and responsibilities, technological requirements that are well defined, organization, a common workflow and concise software stack.
6. **Internal communication:** A common workflow and concise software stack used for issue tracking and version control. Willingness and ability to utilize the recommended tools.
7. **External facilities:** understanding of what facilities have available and what they require, what their schedules look like, how they prioritize university projects, costs involved.

3.3. Maturity levels

The maturity levels can be linked to the factors identified as shown in Fig. 1. This can help plan the next phases, where project managers should identify mitigation actions to reduce the impact of the factors. Furthermore, we recommend using the system readiness levels to focus development efforts. For example, in Fig. 2, we see that **system 1 (SS1)** has an SRL of 0.4, while some of the other NTNU payload software modules are higher (0.6 and above). This means that management should focus resources on developing **SS1** until it reaches a higher maturity level. For hardware, we see the same for **systems 10 and 11 (HS10, HS11)**, which are the environmental interfaces. The project has not yet finalized the thermal subsystem design (HS10), while the payload has been tested in a vibration environment (HS13).

3.4. Interview results

All interviewees (n=5) answered that they would be able to draw a block diagram of their subsystem and its interfaces. The interviewees were able to identify most of the physical interfaces and data links, and some (n=4) identified the interfaces with people within the organization and external (suppliers and support functions at the university). When asked explicitly, the interviewees were able to describe in detail the people and operational interfaces. Some interviewees (n=2) had created ICDs for their documentation, while all interviewees (n=5) have used them for their subsystem development. Most have created readmes for their software, and software team members had created architecture diagrams as well.

People did not use the previous master theses as knowledge source as much as the project documentation. Some interviewees (n=2) introduced the concept of different viewpoints when describing their subsystem, for example the network layers (the 7-layer open systems interconnection model), as one way of describing their subsystem, and which layers they were working on. Most interviewees agreed stated it was important to have the right descriptions of the subsystems used to be able to integrate them properly. For example, how the operating system maps onto the chosen hardware, and that having diagrams showing how the different software modules are interconnected is helpful. How the payload subsystem connects to the interfaces is not supposed to change, but one challenge is to get the right information about the interfaces. It takes time to get the interfaces from third-party suppliers, and one hypothesis is when the technology the team is integrating with is also being developed concurrently to provide the capabilities the mission needs.

Interviewees stated that development was easier when the scope and interfaces were defined. The development challenges were rooted in a lack of specification if not all functions had been identified, and when the architecture decided early in the project did not accommodate the introduction of new functions or capabilities well. It is difficult to know exactly what is needed when the requirements have not been fully developed at the top or derived to specifications. As a young student it was difficult to determine which architecture was suitable for the payload and mission needs. It is easy to start development "too early" without a proper process or guidance.

4. DISCUSSION AND CONCLUSION

The experience from developing the payload identified the following factors influencing the development time: (1) lab facility; (2) university facilities; (3) team knowledge; (4) parts supply chain; (5) clear objectives; (6) internal communication; and (7) external facilities.

The interviewees did not mention *(1) lab*, *(2) university* or *(7) external facilities*, or *(4) parts supply chain* as factors influencing their work. However, these factors are at a managerial or group leader level, and the interviewees were all team members, which may be why these were not mentioned during the interviews.

Regarding *(3) team knowledge*, the interviewees mentioned that it was helpful with all the project documentation such as ICDs, and that there were more senior team members available on Slack to answer questions. However, some interviewees identified that it was challenging to know which questions to ask because it takes time to build the basic knowledge needed. In relation to this, the *(4) parts supply chain* and main subsystem provider influenced the Norwegian University of Science and Technology (NTNU) team more through interface definitions.

The semi-structured interviews showed a strong agreement with *(5) clear objectives*, where well-defined requirements and required capabilities and functions were often not present at the start of a student thesis project. Interviewees identified that development would be easier with well-defined scopes and interfaces, which is linked to having clear objectives of the work. Furthermore, a lack of required capabilities and functions can cause the architecture choice to be incompatible with future refinement of functions and mission.

Finally, some of the interviewees mentioned that *(6) internal communication* was challenging cross-team or cross-function. For example, that the software development was not fully aligned with the operations need. However, the Scrum stand-up meetings help with internal communication, because they provide low level day-to-day tasks and lowers barriers for informal communication.

To enable future planning, we recommend project managers of university CubeSat projects to use SRL, TRL and IRL to help prioritize tasks. It is also a useful communication tool with team members, to ensure alignment and common understanding of the different systems involved. The use of IRL and SRL in addition to the commonly used TRL highlights the need for integration of subsystems which may be forgotten if students are focusing on separate theses. Furthermore, we found that the factors influencing development time vary depending on the maturity level of the technology. For instance, the need for *clear objectives* are critical in the early planning phases and verification phases, or that *university facilities* play a key role in moving from TRL6 to TRL7. The maturity levels can be used for communication with university departments to help plan the workshop and testing availability.

Future work includes using these assessments at multiple gateways of the project, and measure quantitatively how long time each system takes to progress in terms in TRL, IRL and SRL. Furthermore, the authors are also interested in learning about other university CubeSat teams and their experiences with maturity level assessments.

ACKNOWLEDGMENTS

This work was supported by the Norwegian Research Council through the Centre of Autonomous Marine Operations and Systems (NTNU AMOS) (grant no. 223254), the MASSIVE project (grant no. 270959).

REFERENCES

[1] Alminde, L. et al. “Experience and methodology gained from 4 years of student satellite projects”. In: *2nd International Conference on Recent Advances in Space Technologies*. Vol. 1. Turkey, Istanbul: IEEE Xplore, 2005, pp. 94–99.

[2] Richardson, G. et al. “Small Satellite Trends 2009-2013”. In: *AIAA/USU Conference on Small Satellites SSC-VII-3*. 2015.

[3] Cho, M. and Mazui, H. “Best practices for successful lean satellite projects”. In: *4th UNISEC Global Meeting*. Series Best practices for successful lean satellite projects. 2016.

[4] Berthoud, L. et al. “University CubeSat Project Management for Success”. In: *Conference on Small Satellites*. 2019.

[5] Tan, W., Ramirez-Marquez, J., and Sauser, B. “A Probabilistic Approach to System Maturity Assessment”. In: *Systems Engineering* 14 (2011), pp. 279–293.

[6] Mankins, J. *Technology Readiness Level*. Tech. rep. Office of Space Access and Technology, NASA, 1995.

[7] Sauser, B. et al. *A Systems Approach to Expanding the Technology Readiness Level within Defense Acquisition*. Tech. rep. 2009.

[8] Rybicka, J., Tiwari, A., and Leeke, G. A. “Technology readiness level assessment of composites recycling technologies”. In: *Journal of Cleaner Production* 112 (2016), pp. 1001–1012.

[9] Tompkins, Z., Grenn, M., and Roberts, B. “Improving System Maturity Assessments by Incorporating a Design Structure Matrix”. In: *IEEE Transactions on Engineering Management* 67.1 (2020), pp. 122–140.

[10] Tripp, D. “Action research: a methodological introduction”. In: *Educação e Pesquisa* 31 (2005).

[11] Yin, R. K. *Case Study Research and Applications: Design and Methods*. 6th ed. SAGE Publications Inc, 2017.

[12] Austin, M. F. and York, D. M. “System Readiness Assessment (SRA) an Illustrative Example”. In: *Procedia Computer Science* 44 (2015), pp. 486–496.

[13] GitHub, I. *Understanding the GitHub flow · GitHub Guides*. https://guides.github.com/introduction/flow/. (Accessed on 07/03/2020). 2017.

[14] Rising, L. and Janoff, N. S. “The Scrum software development process for small teams”. In: *IEEE Software* 17.4 (2000), pp. 26–32.

Paper D

Honoré-Livermore, Evelyn; Birkeland, Roger, "Managing Product Development and Integration of a University CubeSat in a Locked down World," 2021 IEEE Aerospace Conference (50100), 2021, pp. 1-12, `http://dx.doi.org/10.1109/AERO50100.2021.9438490` [15].

Here, we describe how the COVID-19 pandemic lockdown affected the CubeSat project team, and provide suggestions for project managers to address sociotechnical aspects, schedule impacts, and technical infrastructure.

Managing Product Development and Integration of a University CubeSat in a Locked down World

Evelyn Honoré-Livermore
Department of Electronic Systems
The Norwegian University of Science and Technology
7491 Trondheim
evelyn.livermore@ntnu.no

Roger Birkeland
Department of Electronic Systems
The Norwegian University of Science and Technology
7491 Trondheim
roger.birkeland@ntnu.no

Abstract—**In this article, we describe how digital collaboration tools used during lockdown require new approaches to project management and systems engineering based on the action research case study of a CubeSat student team in Norway. The COVID-19 outbreak affected the whole world during the spring of 2020. Governments reacted by locking down countries and telling people to stay at home, and the university asked the students to work from home. Little previous research was found on managing virtual student teams, especially when not as initially planned. In this situation, many management tasks proved to require more effort than usual, such as managing team members, helping maintain work/home-balance for team members with families or focusing when working from home, and ensuring motivation and on-time project deliveries. The lockdown resulted in an increase of GitHub traffic on the software product. Reasons for this include (a) needing to commit software before the CDR, (b) strengthening of feeling that Github is a platform to work together in when the offices were closed, or (c) maturity of design in general increased contributions to code. All hardware integration efforts were put on hold, but team members expressed that they had time to focus on documentation. In a non-lockdown situation, they would not have done this because of prioritizing "hands-on" work. This may be beneficial in the long run, especially for onboarding new members to the team. Management of a team during lockdown includes evaluating and improving the technical infrastructure necessary for digital collaboration, managing the diversity of situations and other soft issues of a team, and managing schedule impacts both in the short-term and long-term. We found that project managers must make explicit efforts to maintain the project culture and motivation. For example, make efforts to replace the informal interfaces that take place in a co-located team with questions and round-the-table off-topic discussions in stand-ups and meetings. Furthermore, when large changes such as a pandemic happen, it is important to adapt and reinforce the team culture and norms. We also found that having an agile culture made the team more responsive to the change in working norms, such that there was a high willingness to "try out" the best way to lockdown-work.**

TABLE OF CONTENTS

978-1-7281-7436-5/21/$31.00 ©2021 IEEE

1. INTRODUCTION

The recent Coronavirus (COVID-19) pandemic affected almost all countries across the world, and many students found themselves shut out from campus, from seeing their classmates and friends daily, and from traveling home to rejoin their families. Active research and student projects were affected in various ways, such as limited access to laboratory facilities to perform experiments and tests, or reduced mechanical workshop support, or restrictions on field tests. Additionally, the social processes were affected when it was no longer possible to drop by the coffee machine or visit someone's desk during the day for a chat or clarification of questions. Most projects saw the need to use digital collaboration platforms to a higher degree.

The main subject of this paper is to report on experiences of a team of students and researchers who were in the process of completing and testing their CubeSat payload when the pandemic completely reshaped their work environment. For many, distributed teams and digital collaboration are not new. However, when not planned for, the heavy reliance on digital tools can have a significant impact on project progress and team motivation. In global research projects, digital collaboration is sometimes the only means of communication available [1], [2]. Adding to the complexity is the management of a university CubeSat project, where project management, high turnover, transfer of knowledge, and ensuring mission success are recurring issues [3], [4], [5], [6], [7].

This paper is organized as follows. Section 2 provides a review of distributed and dispersed teams and digital collaboration, and introduction to some of the recorded problems CubeSat teams face. Section 3 describes the case study of the Hyper-Spectral SmallSat for Ocean Observation (HYPSO) project and organization, and the method applied for the study. In section 4, a timeline of events is described, including the Critical Design Review (CDR) meeting and results from a follow-up questionnaire. We analyze these events in section 5, with a focus on project management experiences and recommendations. Finally, we summarize the findings and give some lessons learned for managing a CubeSat team and design reviews with digital collaboration tools.

2. BACKGROUND

Distributed Teams and Digital Collaboration

Most organizations and projects have a mix of co-located and distributed or dispersed team members. In this paper, we define a distributed team for where there are both workers on-site in the office and workers in their home offices. A dispersed team is when these workers are spread out over a large area. Both types have significantly been researched in the setting of global teams, where the members can be in different places, time zones, and cultures [1]. Furthermore, team members may be temporarily dispersed in cases of traveling or home offices. The continuous evolution of digital tools has made the logistical aspects of managing distance collaboration simpler to address. However, there are "soft issues" that are important to consider as a manager, such as trust, relationship-building, frequency of communication, and social ties [1], [8], [9]. Most research papers we found discussed *planned* distributed or dispersed teams in businesses, we were not able to find much research on unplanned dispersed teams in academia.

For this team, the goal of the digital collaboration is to share information and to build the satellite collectively. Collaboration can be done by either working together on a task or having individual tasks that are brought together by another team member to deliver the end product [10]. Collaboration requires that there is a shared understanding and acceptance of the goals and deadlines. Furthermore, that the interfaces, both between subsystems, tasks, and people, are understood so that the system can be integrated to meet the goal.

To ensure the success of virtual teams, Kayworth et al. (2000) outlined 14 factors in four categories (communication, culture, technology, project management) needed, based on the case study conducted. These include having norms for communication and culture, training people in using digital collaboration tools, emphasizing continuous contact and communication, understanding the diversity of the team members, and expressing flexibility and empathy [2]. Having flexible project management is also essential for complex and knowledge-intensive projects, such as developing a spacecraft [11].

Verburg et al. (2013) reviewed the aspect of virtuality and types of technology needed to facilitate communication, management and challenges of dispersed teams, and the role of the project manager. Virtuality is related to the frequency and quality of communication and the means of facilitating this communication [1]. In the authors' review, they found that having trust between members and high-quality communication tools are necessary for collaboration. Furthermore, the project manager and organization should have clear rules of etiquette and policies such as muting the microphone when not speaking, or using video when speaking during real-time video meetings.

Virtual team management is enabled by having good organizational support [1], [8], [12]. Drouin et al. (2010) discussed the processes needed for virtual teams such as planning, interpersonal, communication, and collaboration processes. The authors highlighted the need for the team members to be "mindful of the potential differences between their daily reality and that of their foreign collaborators [12, p. 629]." While this case study includes few foreign collaborators, the team members are dispersed throughout Norway and have different daily routines and responsibilities.

A study on predominantly student virtual teams by Panteli et al. (2019) looked at students' experiences and reflections based on a dispersed global team located in United Kingdom (UK) and Norway.The teams studied had not worked together previously, but there was a plan and a structure for how the global virtual collaboration should be with supporting tools. The authors noted that many students had experiences with virtual collaborative platforms through the e-learning platforms, and that some students have used these platforms for teamwork previously. They found that having leadership was critical to success, and to agree to and understand that "norms needed to facilitate virtual team success [10, p. 795]."

The exchange of information and building organizational knowledge is a challenge with co-located teams, which can be more present in dispersed teams [8], [9], [13], [3]. Cramton discussed the types of information problems present in dispersed teams and categorized them into (1) failure to communicate contextual information, (2) difficulties in communicating the salience of information, (3) unevenly distributed information, (4) differences in speed of access to information, and (5) interpreting the meaning of silence [13]. Olaisen and Revang highlight the need for knowledge sharing and knowledge quality, based on the exchange of information combined with "experience, context, interpretation, and reflection [8, p. 1442]."

CubeSats

A CubeSat is a small satellite consisting of cubes, called "units," of 10 cm x 10 cm x 10 cm. The concept was conceived in the early 2000s by professors Puig-Suari and Twiggs [14]. Over the years, the CubeSats have grown from being university educational toys into carriers for versatile scientific instruments and businesses [15]. CubeSats now do a variety of missions, such as communication systems [16] and Earth observation [17]. The size of a CubeSat is diverse. Multiples of units can be combined, for example, into popular sizes of 2, 3, 6, and 12 unit satellites. There is a growing CubeSat community internationally, where businesses, academic, and research institutions together drive the technology onwards [18]. Projects can procure hardware, software, and services from a multitude of providers. Many CubeSat projects combine Commercial-Off-The-Shelf (COTS) subsystems with in-house developed components. Teams can make decisions to buy what they can or make what they need.

The research on project management and systems engineering in academic CubeSat projects highlights issues such as: ensuring continuity when there are high turnover and frequent exam periods [4]; project management and team structure [3], [4], [5], [6]; transfer of knowledge [7]; balancing academic work and satellite building [19] and; ensuring mission success [4], [5].

University CubeSat projects are usually constrained by time and low budget, and are dependent on thesis work by students to complete. There is little or no possibility to engage outside consultants to fill knowledge gaps. Alminde et al. (2005) recommend having Ph.D. students or other long-term resources manage the project, to ensure some continuity and knowledge transfer when the students leave. Furthermore, there must be an explicit effort to "make the students feel like part of a team [4, p. 13]."

Berthoud et al. (2019) conducted an in-depth comparative case study of 3 universities with multiple successful CubeSat missions. Their study showed the importance of having experienced staff leading the initiative, a limited development cycle, passionate students, balance the mix of curricular and extra-curricular work, using version-controlled repositories for managing information and project artifacts, and to emphasize testing. The authors also mention that there are different approaches to formal design reviews and knowledge management, where some have specific processes they follow, while others simply have design reviews and encourage team members to record the design in the various repositories [7].

Ensuring a good design requires broad knowledge and experience. If there is a system and culture for knowledge management, the problem of high turnover is lowered. Furthermore, it is easier for project management to ensure that there is progress in the project and that non-conformances and product development issues are taken seriously. Formal design reviews are used as a tool to facilitate and encourage documentation of knowledge and to motivate project team members and stakeholders to accept a design and its decisions as a whole. Typical design reviews are Mission Design Review (MDR) and Preliminary Design Review (PDR) [20]. These reviews are not as common for academia, where projects are run "ad hoc" towards a prototype or proof-of-concept. However, it is the authors' impression that many CubeSat projects follow *some* systems engineering principles for product development, such as defined milestones or gateways [5], [7]. Additionally, many teams attempt a controlled verification and validation approach to ensure that the satellite conforms to launch vehicle requirements. A recent study [21] highlights how the CubeSat projects classified as "Crafters — characterized by *Streamlined practices, experimentally developed*" have a higher rate of success than "Hobbyists — characterized by *Ad hoc practices*."

3. Case study

CubeSat Team

The CubeSat team at Norwegian University of Science and Technology (NTNU) SmallSat Lab has been developing a mission with a 6U CubeSat since 2017, called HYPSO, with a scheduled (prior to pandemic) launch in Q4-2020. The satellite has a dual purpose: (1) to deliver oceanographic data to end-users (mostly scientists), and to (2) build competence at NTNU to enable fast development of scientific instruments for deployment in CubeSats or autonomous assets.

There is some previous history of building CubeSats and other space engineering products at NTNU, which has resulted in course credits and theses being produced. However, NTNU has had no successful missions to date [22]. The HYPSO CubeSat is the first satellite to be built at the university in recent years, and at the onset there was little knowledge in the faculty about the practicalities involved in building a satellite. The past three years have focused on building both competence and project culture, as well as developing the HyperSpectral Imager (HSI) payload from a Technology Readiness Level (TRL) of 3 to TRL 7.

Organization— The HYPSO project team consists of 5–6 Ph.D. candidates and 20 students writing their Bachelor and Master theses. The team has a project manager and multiple group leaders (all Ph.D. or Post.Doc. students) that are responsible for following up on the design of their subsystems. This includes following up the Bachelor and Master students. The students join the project in either September or January, and most leave in early June. It is challenging to ensure proper transfer of knowledge while simultaneously fulfilling the individual research goals. A thesis does not necessarily include the documentation necessary for someone else to continue the space engineering project. Most of the team members do not have any previous experience with configuration control, formal documentation, project teamwork, or documenting work. However, most of them join the project because they are passionate about space or their specific thesis' tasks. The team members seem motivated to try new techniques and methods to make the project work better.

There has been confusion about design decisions, issues of knowledge management and information flow, lack of clear follow-up and commitment among the project team, and an ad-hoc review process since the project started up in 2017 [3]. In order to improve the project teamwork, it has been suggested to introduce Systems Engineering (SE) and Project Management (PM) methods and tools [23].

Methods of Collaboration— The project team members are encouraged to work on their thesis and the satellite in the lab area, to facilitate concurrent engineering and communication. The main communication channel is **Slack**[1] [24], where day-to-day messages and discussion take place in different topical channels.

No formal processes have been implemented, but a team agreement made in January 2018 stated that all decisions should be documented and that there should be formal reviews and gateways based on tailored European Cooperation for Space Standardization (ECSS) review recommendations. Two reviews had been conducted in the project prior to the introduction of a formal documentation management tool. These have been reported and discussed in [3]. The main findings from this analysis were that the formalized review processes help; using cloud services that support concurrent work on documentation lowers the barriers to contribute to the knowledge base; having a format to provide review comments encourages feedback, but there is a lack of structure and system in the documentation and traceable requirements and follow-up from feedback. The previous reviews have used either questionnaires or spreadsheets to collect feedback.

The digital project management tools have included communication through e-mail, documentation and meeting notes wiki with the possibility of assigning action items, cloud-based spreadsheets and documents, and software code configuration control through **GitHub**[2] using GitHub flow [25]. An automatic Jenkins[3] [26] unit test is run on the master branch of the software on an x86 architecture[4] every night for regression testing of new software changes. The process of software development is shown in Figure 1.

Prior to the events following the COVID-19 outbreak, the team was accustomed to using online meeting tools for regular weekly meetings, ad hoc meetings, and daily stand-ups, as one or more team members often were traveling, working from home or staying abroad.

Method

This paper is based on the method of *action research* through an *exploratory case study* [27]. In action research, the authors themselves are active participants in the project and can influence the team through their behavior. We have chosen the method of exploratory case study because it allows for the discovery of unanticipated behavior and phenomena. The COVID-19 pandemic was an unforeseen event that resulted in a new working environment for the team, which had not been anticipated when the research was started.

The research questions addressed by the paper are:

- **RQ. 1:** What methods and modes of interactions in a university CubeSat research project exist, which methods and

[1] A business communication software
[2] A system for software development version control
[3] Open-source automation server used for automatic testing of software
[4] Instruction set architecture, most computers use this architecture

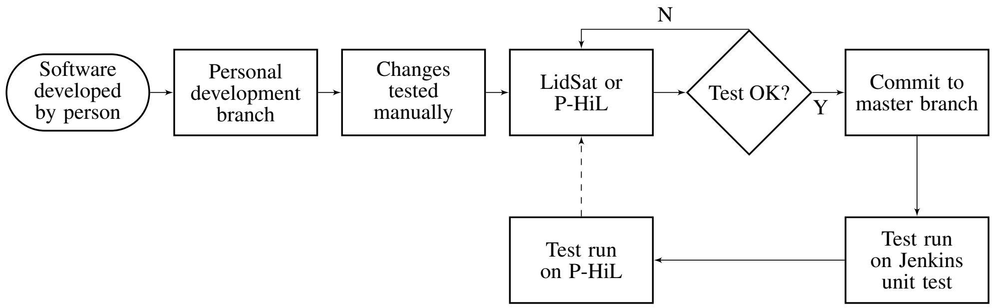


Figure 1: Software development process. Code is exchanged in the flow.

modes work well, and why do they work well?

- **RQ. 2:** How can formalized design reviews improve CubeSat development in a university setting?
- **RQ. 3:** How can project managers run CubeSat research projects successfully when the team is distributed?

The sources of data include (1) digital artifacts in communication channels such as **Slack** and email, (2) reflections from telephone calls and video conferences, (3) feedback from team members through questionnaire, (4) digital artifacts in documentation system, (5) digital design artifacts.

Limitations of the study: The case study is limited to the HYPSO team at the university, which is a small sample size, so that it is difficult to generalize the findings outside this context. The context matters and different countries have taken other measures to the COVID-19 outbreak, so the findings may differ at other CubeSat teams in a different sociotechnical context. Furthermore, as a qualitative study, the findings are based on the interpretations of the authors, which may be interpreted differently by other researchers. However, the findings are in line with other research reviewed, and some generalization and lessons learned can be extracted from the case study and findings.

4. Events

This section will detail the sequence of events and the effects these had on the team and the road to digital collaboration. Figure 2 shows the timeline of national events and for the team.

Hardware-in-the-Loop

While the Jenkins unit testing of the software process provides regression testing of software changes on x86-architecture, it is limited to the unit tests specified by the team. Not all software developed is hardware-agnostic (meaning that it has to run on the target hardware of the payload to function), and it is necessary to have hardware-in-the-loop (HIL) setups to test the software developed.

A HIL-setup was developed and set up starting in January and was almost completed by mid-March. The HIL-setup consists of two systems, called P-HiL (Payload Hardware-in-the-Loop) and LidSat, shown in Figure 3. The P-HiL-setup has been built to enable fast, repeatable, and automated testing of the HYPSO software on the development x86 hardware, which is similar to the target hardware. The purpose of the LidSat is to test the payload software and hardware, in addition to interface with the FlatSat consisting of the other spacecraft subsystems in Vilnius, where the supplier is situated.

Campus Lockdown

The team received the notification that the campus would be closed down on the morning of March 11$^{\text{th}}$. Things happened rather quickly, and some of the team members left campus before the group leaders could talk with them. Many of the students left believing that they would be allowed back on campus in a short time. The first set of regulations asked all students to leave campus. Later on, the regulations were expanded to include all employees who could work from home should work from home. There was an exemption for lab personnel so that essential functions could be maintained.

While some students were developing software, many were working on integration tests in the HIL-setup, and other tasks that require hands-on activities with the hardware. The leaders decided that the students who were working with hardware could bring some copies home with them, although the number of sets was limited. Still believing that the lockdown was short-term, the students were asked to focus on CDR documentation, and their theses work.

Work-from-home adjustment

No students showed up for the first all-digital stand-up, partly because they were not used to using online meeting platforms for stand-up (it used to be in a room), so they forgot about it, and partly because the online meeting platform was not compatible with their operating system. The project manager changed online meeting platforms to find something that worked better with a large team on different operating systems. Considerations included operating system diversity, bandwidth (how the meeting platform could handle changes in bandwidth and adjust the video and audio streams seamlessly), options for interaction, options for viewing. The team members were all favorable to the trialing of multiple platforms until settling on the platform that was most suited to the needs.

Team members reported that it was challenging to manage the interruptions caused by having to take care of kids who could not go to kindergarten or having to share office area with their partner in their homes. Additionally, some reported issues with broadband speed when there were multiple people in their homes needing video-calls or similar. The university also had not prepared for such a demand for their Virtual Private Network (VPN) service and did not have the

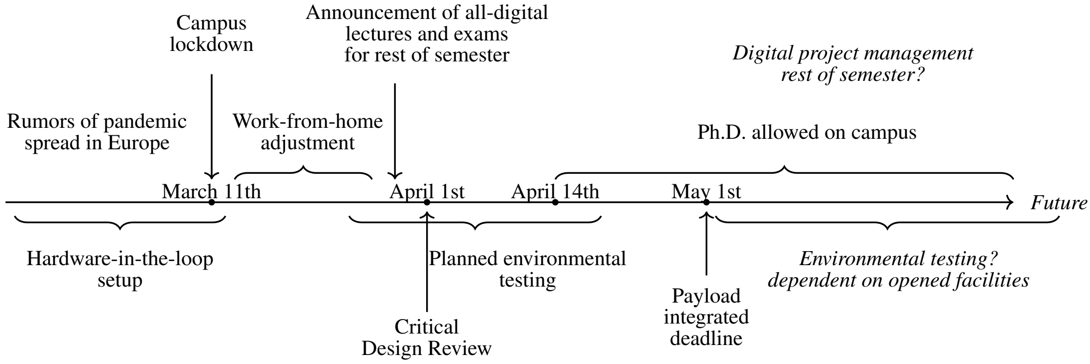


Figure 2: Timeline of events. Text below timeline are planned events, while text above timeline are events related to pandemic and the team's adjustments. Text in italic are open-ended events.

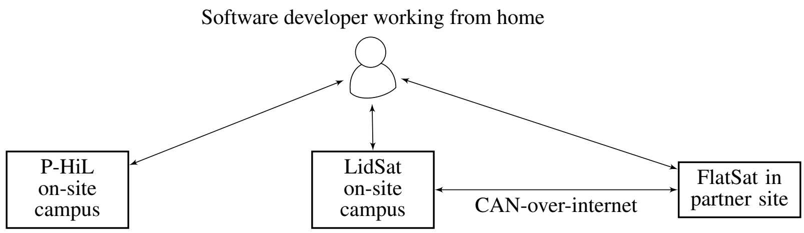


Figure 3: User interface to HIL-setups over internet. CAN=Controlled Area Network protocol used as communication protocol on-board the satellite.

bandwidth to support the requests from everyone at once, so students and employees were encouraged to limit the use of VPN. Some university-resources, including the test and development setups in the lab, are unavailable without a VPN-connection. The capacity of the VPN system was gradually increased over a few days.

Announcement of all-digital lectures and exams for the rest of the semester

At the end of March, the university administration announced that there would be no more scheduled activities such as lectures or exams on campus. All these events should take place digitally or be canceled. For most students, this meant that there was no reason to stay near campus, and some left to go home to stay with their parents instead.

Critical Design Review

The CDR had been planned to happen as a document review process supported by a professional review management tool tailored for the European space sector, ending with a collocation meeting planned to take place on campus. This was changed to being an university-hosted Zoom[5] [28] meeting. The review process was conducted as close as possible to the process recommended by the European Space Agency (ESA) shown in Figure 4. The review team was asked to provide Review Item Discrepancy (RID)s ahead of the collocation meeting so that the HYPSO project team could categorize them into Major and Minor. The collocation meeting focused on discussing and clarifying Major RIDs.

Multiple tools were tested before the design review collocation meeting. The main requirements for the tool to be used were: (1) Good handling of reduced bandwidth with multiple users; (2) Available and functional on multiple operating systems; (3) Possibility for meeting leader to mute/un-mute participants; and (4) Providing participants a *good feeling of belonging*. The last requirement was essential to ensure active participation in the meeting and is dependent on both the meeting leader's inclusion and the tool itself. All tools were tested with more than 10 participants prior to the meeting. In addition to fulfilling requirements (1)-(3), Zoom was chosen because of its "gallery mode" where you could view up to 49 participants in one grid, more than any of the other tools. Furthermore, the possibility to create "breakout rooms" allowed for subsystem discussions during the meeting.

The meeting's duration was from 08:30 to 16:00 on the first day, and from 08:30 to 11:00 on the second day. The first day's agenda started with an introductory round of all participants where they showed video and gave a short introduction. Following that, the mission and project status were presented before a more in-depth presentation of each of the topic areas such as software, hardware, and operations. There were breaks every 90-120 minutes. After the presentations, the participants were divided into five topic groups where the HYPSO team acted as meeting leaders. These groups were "sent" to breakout rooms where they discussed each topic and associated RIDs for approximately 45 minutes, when the review team was "rotated" to go to the next topic

[5] A video and web conferencing tool

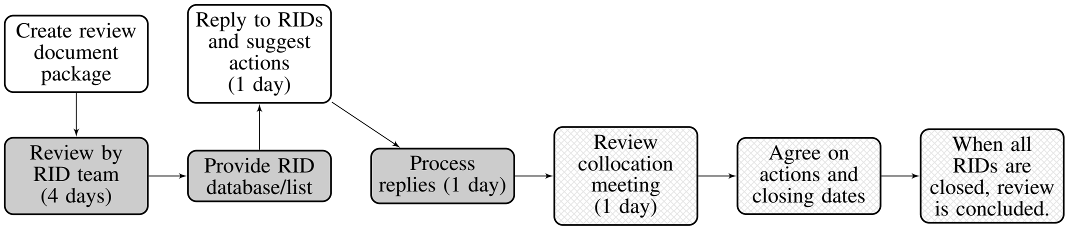


Figure 4: ESA recommended review process. The grey blocks are performed by the review team (RID team), consisting of professors and external reviewers that have experience with space systems, while the white blocks are performed by the project team. The cross-hatched blocks are common.

and associated breakout room. After all the RIDs had been addressed, the HYPSO team agreed on action items and deadlines together with the review team for closing the RIDs. At the end of the first day, the meeting leader asked each topic leader to summarize the findings and described the agenda for the second day.

The second day, the focus was on estimating the probability of meeting the mission and identifying the major unknowns and risks for each topic area. This helped to prioritize the action items. Finally, the meeting concluded with a summary of prioritized areas, and no new meeting was announced.

A voluntary questionnaire was sent out to the participants of the review after the meeting concluded. Out of the approximately 44 participants, 38 responded, $N = 38$ where N equals number of respondents. The total age span was 21-68, and 66.6% of the respondents were in the age group 21-30. This corresponds with most of the HYPSO project team, of which the vast majority are in that age group. The sample size of other age groups was small, and no clear correlation between age and answers could be found in the analysis of the results shown in the paper.

Most of the respondents participated in the meeting using a computer, evenly distributed between Windows, Linux, and Apple operating systems. Some respondents indicated dual-use with a phone as well. The respondents indicated previous experience with different types of video conference tools, including Zoom, as used during CDR. All respondents indicated that they used audio continuously or intermittently, while 26% responded that they did not use video during the meeting.

The meeting was evaluated on a Likert-scale and with an option to add “Other comments” where the results of Likert questions are shown in Figure 5. There is a high degree of agreement that the meeting’s objectives were understood and met, that the meeting stayed on-topic and that the agenda was clear, and that in case RIDs had been submitted, they were answered and things were clarified. More than 50% agree that people spoke less than normally, and that they were more to-the-point in their responses.

The usage of the professional review tool for managing the documentation and review was evaluated on a Likert-scale shown in Figure 6 and with an option to add “Other comments”. There is a strong agreement that the RIDs were helpful, almost 50% responded neutrally if they would use the tool again.

The usage of Zoom for managing the design review was evaluated on a Likert-scale and shown in Figure 7. The respondents were also invited to provide further comments in an “Other comments” open-ended form. More than 90% responded that they would use the tool again and that the option to create breakout rooms was helpful. Close to 95% responded that it was easy to install the tool and join the meeting, as well as being able to hear and see the other participants well.

Planned Environmental Testing

The planned environmental testing at the end of March and the beginning of April did not take place because of national travel restrictions and because of test facilities going into lock-down. Furthermore, some components for finalizing the payload had not been ordered yet, and suppliers were not sure when they could provide them. The planned environmental testing was critical to have enough time to verify that the payload would survive launch and operate in a space environment. The canceling of environmental testing meant that the payload was not fully verified to CDR.

Payload Integrated Deadline

The HYPSO team had planned to have an integrated payload proto-flight model ready for shipping to the spacecraft supplier by May 1st. Because the environmental testing did not happen as planned, this had to be delayed.

Ph.D. Allowed on Campus

The university slowly started opening up for PhD candidates to access the laboratory facilities on a case-by-case basis with limited hourly access. This has enabled resuming some integration activities but with a greater need for planning.

5. Discussion

In this section, three topics of discussion are addressed: (1) Technical infrastructure to enable digital collaboration, (2) Sociotechnical issues and project management, and (3) Schedule impacts.

Technical Infrastructure to Enable Digital Collaboration

This section will discuss some of the tools used to support digital collaboration and how the usage changed, if applicable, during the lockdown and working from home.

The basic challenges of interoperability for tools and access to broadband were most prominent in the first phase (Work-from-home adjustment in Figure 2), and although important to consider when managing distributed teams, will not be

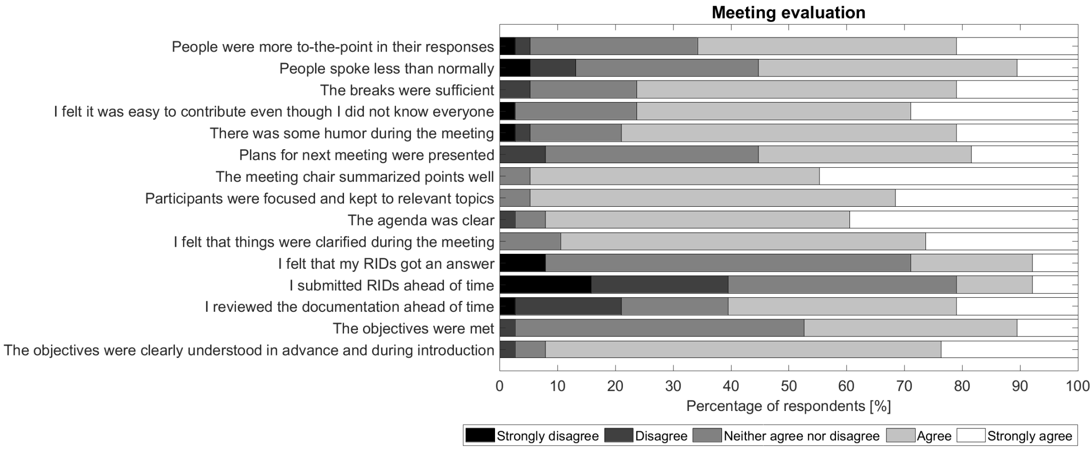


Figure 5: Overall results from meeting impressions.

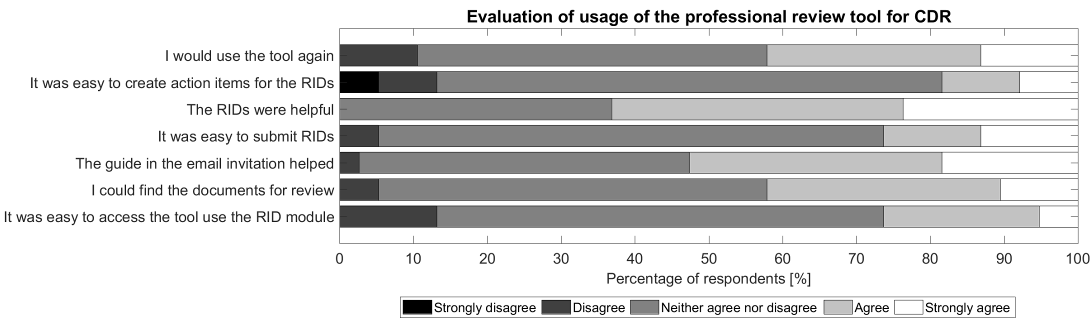


Figure 6: Impression of using the professional review tool for the review and during the meeting.

discussed in detail in this paper.

Software development process. The NTNU team uses GitHub to manage the different software repositories. According to the statistics, the traffic of new software contributions ("new commits") increased after the lockdown, both from existing contributors and with new contributors. There may be several reasons for this, such as (a) needing to commit software before the CDR, (b) strengthening of feeling that Github is a platform to work together in when the offices were closed, or (c) maturity of design in general increased contributions to code. The workflow shown in Figure 1 and GitHub flow [25] are mostly followed.

The main part of the code consists of two individual repositories; one is encompassing the Linux file and operating system for the payload, as well as the Field Programmable Gate Array (FPGA) bit-stream. The second repository holds the rest of the payload software, which is the camera control and processing. Most of the team members contribute to the second repository, while the first has fewer contributors. There have been some problems ensuring alignment of dependencies between the repositories, leading to a non-functioning state of the head of the master branch of the second repository. The root cause was a lack of testing the software changes on feature branches between both repositories before merging them to the master branch[6].

A "broken" master head had not happened before, and the increase in traffic in the repositories increased the likelihood of it happening. At the same time, when the team was all working in the same office, the development was more coordinated because members had continuous informal discussions, so there was a lower likelihood of mismatching between repositories and increased likelihood for testing because the HIL-setups were in the room.

Zoom as a meeting tool. Zoom proved to work well as a platform to execute the CDR. No major technical challenges were encountered (from the organizing perspective). Com-

[6]From GitHub flow: the master branch is the software ready for deployment.

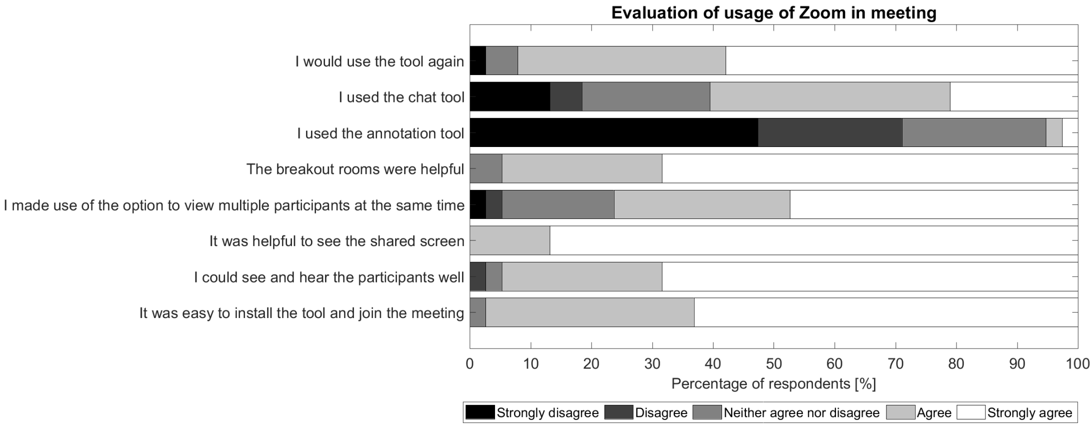


Figure 7: Impression of using Zoom for the meeting.

pared to a physical co-location meeting, the time control was harder to enforce, when presenters went over their allocated time. This was also given as feedback in the questionnaire. During the more interactive breakout room sessions, features such as screen sharing, remote access to other participant's screens, and annotation tool were used. The meeting leader could easily "visit" the smaller groups and move participants between rooms when needed.

Using a professional review tool. The results in Figure 6 summarize that although the tool's functionality is useful, some training is needed to access its functionality. There was no difference in responses in terms of age groups. The tool allows for traceability from documentation to review data package to RIDs to action items that can be assigned to project team members and followed up by the project manager. This adds to the explicit knowledge of the project, which is helpful in CubeSat teams where there is a high turnover, as mentioned earlier. By having design reviews and recording discrepancies and associating action items, the team can find the "paper trail" of task prioritization and design choices. For example, if the version of design lacks a particular analysis, this can be addressed in a RID, which is then associated with the action item "Do thermal analysis on mechanical structure" which in turn can lead to an update in documentation (a new issue or revision). These steps can be traced in the tool, reducing some of the knowledge loss of decisions that can be seen in CubeSat teams [22], [3].

In summary, it is critical to pay attention to the following aspects of technical infrastructure:

- Ensure digital collaboration infrastructure in terms of interoperability.
- Allow time for people to adjust to the changes.
- Allow time for people to become accustomed with using the software development process.
- Use videoconferencing tools where you can see all participants to increase sense of belonging.
- Use breakout rooms in videoconferences to facilitate smaller discussions.
- Use a professional review tool to provide traceability for design and actions, which helps knowledge management in teams with high turnover.

Sociotechnical Issues and Project Management—The technical tools and processes assist the development of the spacecraft, but the team needs to execute these processes and use the tools. We call this a *sociotechnical system* [29]. This section will discuss some of the sociotechnical and project management challenges and give suggestions for best practices.

The main tasks of a project manager are to manage the resources, the schedule, and the scope of work. The project management of a student-based CubeSat team is challenging in itself [4], [5], [6], and in a time of national crisis and high uncertainty, even more so. When the campus lockdown happened, most people believed it would re-open shortly. Both the team members, the group managers, and the project manager were privy to the same information given from the university administration. The uncertainty of the situation made it difficult to plan, and over time the gravity of the pandemic made it clear that it would be a semi-permanent change to how we function as a society.

Project management must enable and support the use of the technical infrastructure and motivate team members to follow the project processes and development flows — the project norms. The project manager must also cultivate the team cohesiveness and culture, and to continue this during the lockdown. The project norms and culture help the team feel connected [1], and these must be maintained and updated when the team becomes distributed. This means communicating the flow (as in Figure 1) and following up frequently with the team members until it becomes a part of the team culture. When there are external changes, such as COVID-19, this should be re-emphasized. We experienced that the large change of the lockdown introduced caused people to partially "reset" the way they worked, and there was a risk of going back to ad hoc practices and not follow the now-established processes and development flows. Furthermore, the project manager must be a champion of using tools for digital collaboration, especially when they are difficult to use, as with the professional review tool, where respondents indicated that the user interface was not optimal [30], [31]. The tool has since been improved with new releases to improve the user interface, and we assume that a new review and survey will give a different feedback. Low overhead and low personal investment are important to ensure that people will take systems into use unless championed by management, or required by management.

We reported on the benefits of being in the same office area in [3], because it allowed for fast closure of issues and increased knowledge exchange and flow of information [13]. While not measured quantitatively, there was a high frequency of informal discussions and conversations when the team was co-located (based on observation study). There were also regular coffee breaks, lunches, dinners, workshops, and birthday celebrations — all to increase the team cohesiveness and lower barriers to encourage interactions. There was an increase of short video-calls or phone-calls in the first few days after the lockdown, because team members had a need to clarify things, and because it was nice to stay in touch with the people whom they were used to seeing on a daily basis. However, some members did not show up for the daily stand-ups, as mentioned earlier. There was also a lack of communication on Slack and e-mail by the same members, and some tasks were getting delayed. While technical difficulties may be one reason, not feeling that they are a part of the distributed team, or that other stress factors make the project work less important and urgent can also be reasons for this. The project manager made sure to continue having stand-ups every day and also reaching out to team members individually to check in and provide feedback. This takes time and requires commitment from the project manager. After the Easter holidays and Labor Day long weekend, the project manager spent a portion of the weekly meetings to talk about non-work topics, which helped fill the gap of the missing informal coffee breaks.

The project manager is also responsible for managing the diversity of the team members, in terms of the diversity of their daily lives and schedules, and the external factors affecting their productivity. This means that the project manager must take an active role in managing and tailoring the tasks to fit the changes caused by the lockdown. For example, by adding resources to developing subsystems or changing deadlines and reducing inter-dependencies of tasks to enable more distributed work [2].

The COVID-19 outbreak caused the team to become distributed, affecting the information flow. On the one hand, when being co-located, people also assumed that knowledge was shared and implicit. On the other hand, when the team is distributed, all information must be made explicit, and people are aware of this, which may be one of the reasons for the increased traffic on GitHub. There was also an increase in the use of discussion channels on Slack, where everything is stored for future reference. While it is too early to conclude, the distributed team structure necessitates more explicit knowledge and having multiple information exchange channels [2]. This can be helpful for off-and on-boarding when the current project team leaves at the end of a semester, and a new one arrives after summer. If the lockdown continues, it will be interesting to see how to successfully on-board a fully digital distributed team.

The risk management process of the project had not identified being locked out of campus as a potential risk, neither in the short-term nor the long-term. The country had not been strongly affected by the previous recent pandemics (Severe Acute Respiratory Syndrome (SARS) in 2003 or Middle-Eastern Respiratory Syndrome (MERS) in 2015), and while a pandemic had been identified as a risk by the Norwegian Directorate for Civil Protection [32], this was not included in the project risk assessment. Although it cannot be expected that a university-based project should take large national risks into account. At the sime time, some of the pandemic impact could have been identified with corresponding mitigation actions. The risk acceptance and risk understanding impact the risk management. As a society on a macro level, and as the project team on a micro level, there might have been a risk aversion against a pandemic since we did not take it into account. Furthermore, we did not fully understand the risk and what it would mean to the project and our daily lives. The project risk management is conducted in the professional review tool and some of the impacts such as "R-1: delays in supply chain", "R-2: lack of access to testing facilities", "R-3: team members not present in on-site" or "R-4: HIL-setup not available" were addressed as separate risks (R-1 and R-3), but the risk of all of them happening at the same time was not handled which is an avenue for future improvement for university CubeSat teams. Furthermore, R-1 was assessed to have a low likelihood since the supply chain was mostly in-house, and the risk of the internal workshop not being available was not considered. Similarly, R-3 had been assessed as low impact and low likelihood because some measures had been taken (regular meetings with videoconferencing tools, the commitment of team members to be on campus), and it seemed probable that the team members would be present on campus since that is regular mode of operation for the project members. This can, in part, be attributed to *availability bias* or the *availability heuristics*, where our risk assessment is influenced by how available our memory of events is. For example, if something has happened recently that had a strong impact, even for a short period, we would remember it vividly and might assess it as having a high probability and impact — even though it objectively did not. While the virus spread in China brought up in the weekly meetings prior to the campus lockdown, we did not consider that we might be asked to work from home. However, given that Norway had not been strongly affected by the previous pandemics, our biases downplayed the risk likelihood and impacts. In the case of R-3, because it had not been a problem lately, we assessed it as low probability and low impact because we could not imagine it happening.

For project managers, we highlight the following lessons learned:

- Make an explicit effort to replace the informal interfaces that take place in a co-located team with questions and round-the-table off-topic discussions in stand-ups and meetings.
- Distributed team members have diverse daily lives, and this diversity must be managed by being flexible and by regular individual communication with team members.
- It takes more time and commitment from the project manager to follow up distributed team members than when co-located.
- When large changes such as a pandemic happen, it is important to adapt and reinforce the team culture and norms.
- Having an agile and flexible mindset can help mitigate the impacts of risks not identified.
- Cognitive bias' will affect the risk acceptance and risk understanding of team members, and knowing this helps to manage the risks.

Schedule Impacts

The COVID-19 outbreak caused an overall delay to the project, partly because of lower productivity, responsiveness in the supply chain, and because of the availability of testing facilities. This section will discuss some of the main issues and impacts on schedule, and some suggestions for managing the schedule of a university CubeSat project based on the experiences had.

There were both short-term and long-term impacts on the schedule from the COVID-19 outbreak. In the short term, it meant little or no accessibility to testing facilities, and slower delivery of parts because the supply chain was operating at lower speed due to various restrictions. Longer-term effects come further down in the supply chain, such as deliveries of components, electronics, raw materials for machining, and less dependable freight of components. A lesson learned from the supply chain management, also applicable to university student projects that cannot afford the long delays that larger corporations might have funds for, is to manage the risk of having a *Just-in-Time* (JIT) supply culture, which is strongly affected by pandemics [33], [34].

The delay of delivering a verified and integrated payload meant that the timeline for payload-spacecraft integration must be shortened, increasing the risk of the mission. Furthermore, a new approach with remote functional testing had to be planned in case the team members could not travel to the testing facilities. This added more work to the software development team, which was challenging when most of the team members were supposed to reduce their project work and focus on writing their master theses.

Project managers of university CubeSat projects should consider the following:

- Continuously managing the project schedule and communicating it to the team — making sure that there is an understanding of interdependencies of tasks.
- Be flexible and re-prioritize tasks when the students go into thesis writing phase — focus on few and well-defined deliverables for project.
- Work together with mission responsible to adapt mission objectives to delays in development causing a more compact schedule.
- Consider the JIT supply culture of the world and plan for mission adaption if components and functions cannot be delivered on time.

6. CONCLUSION

The findings from this study provide some best practices and insights that can be useful for distributed and dispersed engineering teams. While the case study focused on an academic context, the results and recommendations are relevant for industry professions also. Especially, we want to highlight the importance of project managers to manage the sociotechnical aspects of teamworks, ensuring that practices and culture are cultivated, and paying attention to managing diverse situations in a lockdown. The study looked at both normal project work and at conducting project milestone, the Critical Design Review. We found that when building a hardware-software system, in this case, a CubeSat, it is important to consider the following aspects: (1) technical infrastructure to enable digital collaboration and ease of working-from-home; (2) sociotechnical issues regarding collaboration and team cohesiveness are prone to influence by factors outside the project; (3) and compound schedule impacts forced by the pandemic may be present in future projects and need to be addressed by the project and risk management.

Addressing research questions. In response to RQ. 1, we found that there are multiple modes of interaction present in a university CubeSat team, both formal and informal modes. The formal ones include regular stand-up meetings and group meetings, while informal ones are communication on Slack, e-mail, on GitHub, coffee breaks, in the co-located working area, etc. After the lock-down, it became clear that the importance of Slack and GitHub increased, and that these also function as information flow channels [13]. Furthermore, including more off-topic and informal discussions in formal meetings should be facilitated by the project manager, to build a digital informal communication culture to replace the co-located communication channels we used to have. These interaction methods and modes work well because they each have their purpose — Slack for clarifying and discussing architectural or system topics, GitHub for specific software issues, meetings for larger "face-to-face" discussions that end up documented in GitHub or meeting notes. This is in agreement with Kayworth et al. (2000) which identified having multiple communication channels for different purposes as a success factor for virtual teams [2].

Having formalized design reviews (RQ. 2) helps ensure the traceability of design and design decisions. They also act as a team cohesiveness event in the way that there is a common understanding of the design and the way forward to reach the project goals and objectives. Usage of the professional review tool greatly helped documenting the traceability from design to actions and back to design again, and makes it is easier to manage the design knowledge for future team members. By this, we propose that having professional tools can be helpful for CubeSat teams, but that they require a champion and support from project management to be utilized and maintained.

In response to RQ. 3, these are some lessons learned for project managers of CubeSat projects in a distributed team setting. Firstly, it is important to choose digital collaboration tools and establish norms that fit the project phase and people, and to champion these continuously. For example, introduce work-from-home options and the collaborations tools even if there is no lockdown, to accommodate working parents, or teachers on sabbatical. Secondly, as a project is a sociotechnical system where people and processes matter, the project manager's most important task is to balance progress with empathy for individual situations. For university CubeSat projects, this means understanding the diversity in the daily lives where many students may be all alone in the lockdown while others also have families they need to be with. Project managers need to make explicit efforts to maintain the team culture and facilitate the informal communication that happens naturally when co-located. Thirdly, project managers should work with the persons responsible for adapting mission objectives to fit the changing development schedule and tailor the schedule and project tasks for students to fit their curricular obligations (such as thesis writing). In summary, having a flexible or agile, project management is key to successfully leading distributed teams [2].

APPENDIX

List of Software Tools Used

- **GitHub**: A system for software development version control
- **Slack**: A business communication software
- **Jenkins**: Open-source automation server used for automatic testing of software

ACKNOWLEDGMENTS

This work is supported by the Norwegian Research Council (Grant No. 270959), the Norwegian Space Agency, and the Centre of Autonomous Marine Operations and Systems (NTNU AMOS).

REFERENCES

[1] R. M. Verburg, P. Bosch-Sijtsema, and M. Vartiainen, "Getting it done: Critical success factors for project managers in virtual work settings," *International Journal of Project Management*, vol. 31, no. 1, pp. 68–79, 2013.

[2] T. Kayworth and D. Leidner, "The global virtual manager: a prescription for success," *European Management Journal*, vol. 18, no. 2, pp. 183–194, 2000.

[3] E. Honoré-Livermore, "Cubesats in university: Using systems engineering tools to improve reviews and knowledge management," *Procedia Computer Science*, vol. 153, pp. 63–70, 2019.

[4] L. Alminde, M. Bisgaard, D. D. V. Bhanderi, and J. D. Nielsen, "Experience and methodology gained from 4 years of student satellite projects," in *2nd International Conference on Recent Advances in Space Technologies*, vol. 1. Turkey, Istanbul: IEEE Xplore, 2005, pp. 94–99.

[5] M. Cho, "Best practices for successful lean satellite projects," 2016. [Online]. Available: https://unisec2016.castra.org/index.php/unisec2016/unisec2016/paper/view/49/10

[6] J. Bouwmeester, G. Aalbers, and W. Ubbels, "Preliminary mission results and project evaluation of the delfi-c3 nano-satellite," in *4S Symposium Small Satellites Systems and Services*, 2008.

[7] L. Berthoud, M. Swartout, J. Cutler, D. Klumpar, J. A. Larsen, and J. D. Nielsen, "University cubesat project management for success," in *Conference on Small Satellites*, 2019.

[8] J. Olaisen and O. Revang, "Working smarter and greener: Collaborative knowledge sharing in virtual global project teams," *International Journal of Information Management*, vol. 37, no. 1, Part A, pp. 1441–1448, 2017.

[9] S. Jarvenpaa and D. Leidner, "Communication and trust in global virtual teams," *J. Computer-Mediated Communication*, vol. 3, 1998.

[10] N. Panteli, T. A. Hjeltnes, and K. A. Strand, "Learning to lead online collaborations: Insights from student-based global virtual teams between uk and norway," in *18th International Federation of Information Processing (IFIP) Conference on e-Business, e-Services, and e-Society (I3E)*, I. O. Pappas, P. Mikalef, L. Jaccheri, Y. K. Dwivedi, M. Mäntymäki, and J. Krogstie, Eds., Trondheim, Norway, 2019, pp. 785–796.

[11] R. Dove and B. Schindel, "Agile systems engineering life cycle model for mixed discipline engineering," in *INCOSE International Symposium 2019*, Orlando, Florida, 2019.

[12] N. Drouin, M. Bourgault, and C. Gervais, "Effects of organizational support on components of virtual project teams," *International Journal of Managing Projects in Business*, vol. 3, pp. 625–641, 2010.

[13] C. Cramton, "Information problems in dispersed teams," in *Academy of Management Proceedings*, 1997.

[14] H. Heidt *et al.*, "CubeSat: A New Generation of Picosatellite for Education and Industry Low-Cost Space Experimentation ," in *Proceedings of the AIAA/USU Conference on Small Satellites, SSC00-V-5*, 2000. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2000/All2000/32/

[15] M. N. Sweeting, "Modern small satellites-changing the economics of space," *Proceedings of the IEEE*, vol. 106, no. 3, pp. 343–361, March 2018.

[16] Kepler, "Kepler's first ku-band satellite is in orbit," accessed February 2018. [Online]. Available: http://www.keplercommunications.com/blog/item/first-launch

[17] Planet, "Planet webpage," 2016, accessed April 2016. [Online]. Available: www.planet.com

[18] Small Spacecraft Systems Virtual Institute, "State of the art - small spacecraft technology," Ames Research Center, Moffett Field, California , Tech. Rep. NASA/TP–2015–216648/REV1, 2018. [Online]. Available: https://www.nasa.gov/sites/default/files/atoms/files/soa2018_final_doc-6.pdf

[19] C. D. Abadi and T. Bahill, "Product and process: an innovative approach for a student satellite program from the systems engineering point of view," in *15th Annual/USU Conference on Small Satellites*, 2001.

[20] Requirements and E.-E. Standards Division, "Space management: Organization and conduct of reviews," 2008.

[21] M. Swartout, "Cubesat mission success: Are we getting better?" in *2019 CubeSat Developers' Workshop*, 2019, Conference Paper.

[22] J. Grande, R. Birkeland, A. Gjersvik, S. Mathisen, and C. Stausland, "Norwegian student satellite program – lessons learned," in *68th International Astronautical Congress*, 2017, Conference Paper.

[23] P. Waswa and A. M. Madni, "A survey of space mission architecture and system actualization methodologies," *International Journal of Space Science and Engineering*, vol. 4, no. 3, 2017.

[24] Slack Technologies, *Slack*, San Francisco, CA, USA, 2020. [Online]. Available: https://www.slack.com

[25] GitHub, "Understanding the github flow · github guides," https://guides.github.com/introduction/flow/, (Accessed on 04/23/2020).

[26] K. Kawaguchi, *Jenkins software*, Software in the Public Interest, USA, 2020. [Online]. Available: https://www.jenkins.io/

[27] R. K. Yin, *Case Study Research and Applications: Design and Methods*, 6th ed. SAGE Publications Inc, 2017.

[28] Zoom Video Communications, Inc., *Zoom*, San Jose, CA, USA, 2020. [Online]. Available: https://www.zoom.us

[29] W. B. Rouse, M. J. Pennock, and J. Cardoso, "Advances in sociotechnical systems," in *Advances in Systems Engineering*, J. Hsu and R. Curran, Eds. American Institute of Aeronautics and Astronautucs, Inc., 2016, pp. 35–64.

[30] G. H. A. Morton, "Become a project champion," *International Journal of Project Management*, vol. 1, no. 4, pp. 197–203, 1983.

[31] J. Pinto and D. P. Slevin, "The project champion: key to implementation success," *Project Management Journal*, vol. 20, pp. 15–20, 1989.

[32] Direktoratet for samfunnssikkerthet og beredskap, "Analyse av krisescenarioer 2019," Tech. Rep., 2019.

[33] L. J. Radonovich, P. D. Magalian, M. K. Hollingsworth, and G. Baracco, "Stockpiling supplies for the next influenza pandemic," *Emerging infectious diseases*, vol. 15, no. 6, pp. e1–e1, 2009.

[34] M. T. Osterholm, "Preparing for the next pandemic," *N Engl J Med*, vol. 352, no. 18, pp. 1839–42, 2005.

BIOGRAPHY

Evelyn Honoré-Livermore *received her M.Sc. in Electronic Engineering at NTNU and her MBA from Yonsei University in Seoul. She is a Ph.D. Fellow at NTNU in the Department of Electronic Systems. She is researching systems engineering and project management methods for academic research projects. She is also the project manager of the small satellite HYPSO.*

Roger Birkeland *received his M.Sc. in Electronic Engineering at NTNU and is a post-doctoral researcher at NTNU in the Department of Electronic Systems. He received his Ph.D. in satellite communications in (2019) and is currently researching small satellite systems and heterogeneous communication systems for remote areas.*

Paper E

Honoré-Livermore, Evelyn; Birkeland, Roger; Bakken, Sivert; Garrett, Joseph L.; Haskins, Cecilia, "Digital Engineering Management in an Academic CubeSat Project," Special Issue on Systems Engineering Challenges in Journal of Aerospace Information Systems (2021) [16]. https://doi.org/10.2514/1.I010981.

In this paper we report on how a digital engineering approach has contributed to improving the working environment and development processes of a university CubeSat team.

Digital Engineering Development in an Academic CubeSat Project

Evelyn Honoré-Livermore *, Roger Birkeland†, Sivert Bakken‡, Joseph L. Garrett§, and Cecilia Haskins¶
Norwegian University of Science and Technology, Trondheim, Norway

Digital engineering is increasingly introduced for managing and supporting the development of systems for space. However, few academic teams have the competency needed to manage projects using digital engineering and systems engineering. The subject of this paper is an academic CubeSat project in which a variety of digital engineering techniques are used. The tailoring that has been applied to fit the academic environment including students from different disciplines and levels of maturity is described. We show how a customized Scrum methodology for hardware and software integrated with a workflow in a digital tool environment has given positive results for both the team and the system development. We also discuss how to introduce new members to the team and how to train them to work with digital engineering as a multi-disciplinary team. We present how the systems engineering and project management activities have been integrated into the academic CubeSat project, evaluate how well this fusion worked, and estimate its potential to be used as a guide for other digital engineering projects.

I. Introduction

The digital transformation that is taking place in all elements of society calls for continuously updated knowledge for leaders and for engineers. The increasing project complexity introduced by the advent of embedded systems and Cyber-Physical Systems (CPS), and the tools needed for developing them challenges managers to re-think the approach to leading projects and people to ensure knowledge management and project success [1]. While this is challenging in industrial settings with experienced engineers and support systems, developing complex systems in an academic environment adds factors such as high turnover, coursework, lack of multidisciplinary teamwork experience, and fewer competent Systems Engineering (SE) and Project Management (PM) resources.

Digital engineering and Model-Based Systems Engineering (MBSE) are proposed as tools to manage the challenges of developing systems, delivering integrated multidisciplinary product development from concept through the product

*Ph.D. Candidate, Department of Electronic Systems, evelyn.livermore@ntnu.no, and AIAA student member.
†Post-doctoral researcher, Department of Electronic Systems, roger.birkeland@ntnu.no
‡Ph.D. Candidate, Department of Engineering Cybernetics, sivert.bakken@ntnu.no
§Post-doctoral researcher, Department of Engineering Cybernetics, joseph.garrett@ntnu.no
¶Associate Professor, Department of Mechanical and Industrial Engineering, cecilia.haskins@ntnu.no

life-cycle to retirement. We adopt the Digital Engineering (DE) definition of the U.S. Department of Defense (DoD): "an integrated digital approach that uses authoritative sources of system data and models and a continuum across disciplines to support lifecycle activities from concept through disposal" [2, p. 340]. For MBSE, we use the definition provided by International Council of Systems Engineering (INCOSE): "The formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases" [3]. However, choosing the approach tools and methods to introduce and adopt DE is equally challenging and requires both human and technical resources.

Concurrent with the advent of digital engineering, approaches such as Scrum and Extreme Programming (XP) have increased in popularity both for hardware and software [4]. The Scrum methodology allows for agile product development, so that the project can respond to changing demands from stakeholders and new technology developments while continuously delivering features. The digital Scrum tools also provide a system which support *project management* through feature and schedule management, *product management* through scope and verification management, and may be integrated with the digital design artifacts. Extreme Programming takes iterative development to an "extreme" level, with short iterations, continuous test development, pair programming, continuous integration, and frequent releases [5]. In software projects where there is scientific code development, and requirements are either unknown at the beginning or frequently change, XP or Scrum are suitable over other traditional approaches [6].

Students in academic projects face the challenge of balancing coursework and project work. The students follow the school-year, so long-term academic projects must adapt their expectations to this fluctuation and there is a high natural turnover the team composition when students graduate. Academic projects may have fewer resources and fewer support systems that product development often necessitates (e.g. a procurement department or quality assurance knowledge) [7, 8]. The university context requires attention to knowledge transfer and management, and digital engineering is a tool that can be applied and must be managed to enable a good development environment.

This paper is based on the longitudinal case study of an academic CubeSat where the students typically join in September and leave in June the following year, although some students join in January and leave in June the same year. They contribute to the development of the CubeSat through work toward a thesis in either software, hardware, or theoretical studies. We explore the cycle of development of a CubeSat in an academic environment using digital engineering tools and describe how they have been tailored. Furthermore, we discuss how MBSE has been applied and what barriers for use of were experienced. We found that using agile practices powered with DE tools and processes greatly improved information sharing and knowledge management, and that the introduction of remotely accessible hardware-in-the-loop (HIL) setups coupled with a defined workflow enabled improved verification, validation, and integration activities.

II. Background

A. Academic CubeSat projects

Since the definition of the Cube Satellite (CubeSat) standard around the year 2000, applied space technology and satellite production has become a staple offering at universities [9]. At first, most initial CubeSat projects sought to evaluate the viability of CubeSats as a concept, and limited their initial goal to communication. Over the last 20 years, the missions have evolved in sophistication into projects with more advanced research objectives [10]. To meet the needs of this burgeoning industry, a substantial supply chain for CubeSat buses and subsystems has been established so that university researchers can then focus upon their main task: defining and building the payload and without having to build the rest of the spacecraft bus around the payload too. In most cases this saves both cost and development time.

CubeSats are built from units (U) of $10\,\mathrm{cm} \times 10\,\mathrm{cm} \times 10\,\mathrm{cm}$, ranging from 0.25U to 16U, with 3U being the most common size [11]. Larger satellites at 6U and 12U are becoming increasingly popular. As the technology matures, the satellites' capabilities increase, for example including advanced deployable mechanisms for solar panels and instruments. With this maturity the missions are becoming more advanced and can deliver more valuable results.

The lifecycle of an academic CubeSat project typically starts with an idea for a research project or an educational CubeSat, then securing the funding, moving on to the preliminary design phase, the critical design phase, launch of the CubeSat (when funding is available). Then follows the operational phase with payload data collection and analysis (if successful), and finally decommissioning at the end of spacecraft lifetime. This takes from 1–5 years, with an average of 3.8 years [12].

The CubeSat subsystems are usually highly integrated, and modularity is ensured both in software and hardware [13]. As the cost of fixing problems increases later in the development cycle, during integration, testing, and maintenance [14], early integration and testing are encouraged. To a large degree, the subsystems can be considered a cyber-physical system because their performance depends on both the hardware and software developed. The integration process can be improved by using advanced, industrial-type electronics and computational platforms during development and test, the integration process can be improved. Using as many Commercial-Off-The-Shelf (COTS) components as possible, lead-time is reduced, and development can be based upon well-known tools with little or no adaptation. There is an opportunity to reduce the risk of late discovery of bugs by proactively using HIL setups throughout the development cycle, enabling iterative development.

Opportunities for education and training using CubeSats To date, over 400 university satellites have been launched, with more than 500 in the pipeline [11]. The educational benefit and the use of CubeSat programs as an introduction to applied space technology has been much discussed in the CubeSat community [7, 15–17]. The first educational CubeSats provided students an opportunity to follow a space project from start to launch within their time at a university. Hands-on projects give students a realistic, but manageable "first contact" with space projects and space industry [18]. Institutional actors such as National Aeronautics and Space Administration (NASA) and European Space Agency (ESA) promote and support educational CubeSats by enabling contact and access to space professionals, and by facilitating courses and workshops as well as launch for the best qualified satellites through their ELaNa [19] and Fly-your-satellite [20] programs. This applied work also motivated many university teams to create spin-offs from their projects, becoming central players in the CubeSat community and a part of the supply chain. They now form a substantial ecosystem where it is possible to procure everything from single components to a turn-key mission where you define your payload and the satellite provider does the rest.

B. Agile methodology and development practices

Using agile methodologies in software and hardware development has gained popularity in the past decades, focusing on continuous feedback from the customer and the ability to react to a changing environment [21, 22]. The word "agile" has its etymological source from the Latin word *agilis*, which means "can be moved easily, light", and from the French word *agere*, which means "to drive, to be in motion" [23]. In software development, the agile methodology gained popularity in the late 1990s, and "Manifesto for Agile Software Development" [24] with its 12 guiding principles was published in 2001. The manifesto includes principles that focus on delivering the highest value to the customer, to allow for changing requirements, frequent and iterative deliveries of software, motivating individuals, face-to-face conversations, measuring progress through working software products, simplicity, reflexive practices, and believing that the best designs come from self-organizing teams [24].

At universities, software and hardware development serve both to assist scientists in gathering data, and for teaching technology and product development. In most cases, the development is not done with the purpose of delivering a mass-produced product or service, but for the purpose of contributing to new knowledge and research. A key challenge of scientific software development is that the scientists often have formal education in a field other than computer science, for example in biology, remote sensing, electronics, or radio technology, but need custom software to address their discipline-specific research questions [25, 26]. Given the open-ended nature of research projects, the process of requirements specification lacks maturity in comparison to industrial development projects, making it challenging to plan the development and to test the software. Furthermore, the scientific software development does not "stop" when the first research project ends, but it may be reused in a different research project with different goals, and new scientists desiring new functionality [27].

Best practices for scientific software development include: write programs so that the other researchers understand and stick to a code style and formatting, make the frequently used commands easily accessible, incremental development with continuous testing, use version control, "plan for mistakes" and use unit testing, improve performance after the functionality is there, document the design and interfaces, and choices made during development, and collaborate on code development and do code reviews [25]. Typical challenges facing scientific software teams are "compromising between feature demands and quality control; code ownership and management during evolution; data organisation and curation; and quality assurance of heterogeneous components, (...) and a tendency for prototyping practices to be employed even when production scientific software was being written [28, p. 47:6-7]." In Arvanitou et al., software practices for scientific development were discussed based on an extensive literature study [26]. They found that most scientific software engineering literature has studied process improvement, ease of development, testing and verification, project management, coding and quality assurance. Furthermore, that *performance*, *maintainability* and *development productivity* were the highest priorities for the scientists.

In a survey of agile methods in scientific programming in disciplines such as bioinformatics, climate scientists, and aerospace, it was found that the agile method XP has been applied successfully in projects where requirements and design cannot be known in the planning phase of a project [28]. Furthermore, agile practices such as iterative development, continuous integration, and version control, were prominent. In contrast to commercial and industrial software development, there is no declared or identified customer to review the software features. However, scientific publications can be analogous to customers in which the scientists receive feedback on what they have developed [28, 29]. Sletholt et al. [27] conducted a literature review against 35 agile practices from Scrum and XP, and found some support that agile practices are suited to testing-related activities.

Agile practices in teaching have gained popularity since the 2000s [30, 31], where Scrum or XP have been the most prominent methods, and typically found in either software or capstone projects. The students benefit from learning hands-on project experience, learning to prioritize work tasks, gaining communication skills, and providing and receiving assessment on work done openly. However, there may be challenges in terms of balancing time commitments, for example having concurrent development sessions, or tailoring the Scrum processes to suit the different needs of team members [31]. Lundqvist et al. [32] reported on teaching agile in cooperation with industry. They highlighted the importance of ownership, the engagement of customer, also called the industrial partner, and the allocation of academic resources to support the academic teams.

According to a study from Australia in 2015, employers want both technical skills and non-technical professional skills such as "being able to communicate effectively," "ability to organise work and manage time effectively," "being willing to face and learn from errors and listen openly to feedback," "being able to empathise with and work productively with people from a wide range of backgrounds" [33, pp. 263–264]. A similar study conducted in Norway also highlighted these points [34]. However, the traditional form of classroom teaching may not facilitate the development of these skills effectively. Using CubeSats for training students in cross-disciplinary projects has been studied and discussed [7, 12, 15–17]. Some principles for agile SE that have been suggested include (1) focus on delivering customer value, (2) team ownership, (3) embrace change, (4) continuous integration, (5) test-driven, and (6) taking a scientific approach to systems' thinking [29, 35]. Many of these principles are aligned with transferable skills students can be expected to have when they graduate [33, 34].

C. Digital engineering

Digital engineering goes beyond "just" using computer tools to aid engineering, but includes the engineering process and approach to development. Choosing a DE strategy should be done based on the resources available and needs of the organization. A framework that assesses the DE competence was developed by the Systems Engineering Research Center (SERC) which looked at the following areas: *adoption*, *velocity/agility*, *knowledge transfer*, *user interface*, and *quality* [36]. While the framework did not specify how to measure the competence in each of the areas, it listed different factors and examples of processes or outcome metrics that could be used. Some factors identified can be categorized as objectives for why DE measures are incorporated, others as factors which may influence the adoption, and other factors as outcomes and direct competencies the organization can gain with DE practices. DE has a strong relationship with MBSE and Model-Centric Engineering (MCE), and establishing a "single source of truth" for a project [2]. However, there is currently no single solution for the whole system lifecycle to provide an authoritative source of truth. Most work-forces and organizations need to transition their methods and methodologies to DE and incorporate it into their engineering practices, and ensure possibilities for collaboration and information sharing throughout the system lifecycle between developers and the stakeholders. Most university CubeSat teams use some degree of DE, such as employing version-controlled software repositories, using CAD tools, shared cloud documentation, and using cloud-based issue tracking or project management tools to achieve integration in the management of knowledge [12].

Garzaniti et al. [37] also describe the use of Scrum using an online tool to manage the work in an academic CubeSat team. The results presented were from the preliminary design phase of the space hardware. They found that the Scrum approach was helpful for reacting to unforeseen changes and delays, even when the changes impacted external manufacturers. Furthermore, that it takes time for the team to become accustomed to Scrum and the scoring of issues, similar to [31]. Huang et al. [38] describe the development of a CubeSat using agile practices. They highlight the importance of tailoring the approach to the needs of the project, using interactive design reviews to produce as much feedback as possible, empowering smaller teams to enable faster decision-making and ownership, and allowing for continuous testing and improvement.

III. The HYPSO case study

A. The HYPSO CubeSat project

In this paper we report on the case study of the CubeSat project Hyper-Spectral SmallSat for Ocean Observation (HYPSO). It is the first research CubeSat mission for the Norwegian University of Science and Technology (NTNU), as a part of a strategy of monitoring coastal areas using autonomous assets [39]. The project's mission is to:

> *"To provide and support ocean color mapping through a Hyperspectral Imager (HSI) payload, autonomously processed data, and on-demand autonomous communications in a concert of robotic agents at the Norwegian*

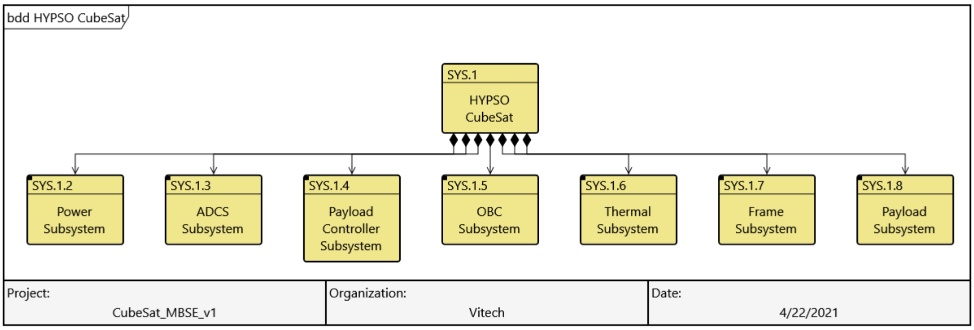


Fig. 1 Overview of the HYPSO CubeSat and its subsystems. Model made using CORE/GENESYS.

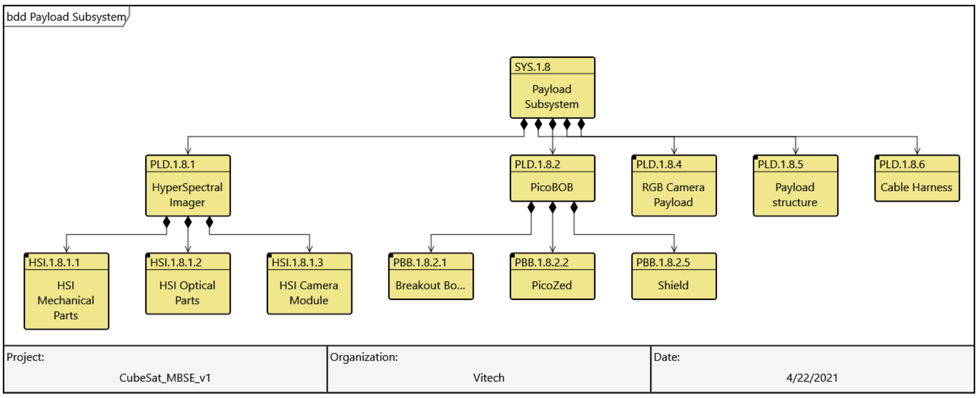


Fig. 2 Overview of the payload developed by the HYPSO team. Model made using CORE/GENESYS.

coast.”

The university CubeSat team develops the payload, which consists of an optical telescope, a COTS camera unit, a COTS processing unit, an electronics interface board, an electrical harness, software to control the payload and to perform the image processing, and mechanical support structure which also acts as the mechanical interface to the satellite bus. Block diagrams of the spacecraft and the payload are given in Fig. 1 and Fig. 2, respectively. Apart from the above-mentioned COTS components, all have been developed in-house. In addition to the payload, there is also development of a local ground station and the mission operations center and associated procedures and functionality, effectively resulting in a System of Systems (SoS).

The CubeSat project team includes 10–20 MSc and BSc students, one electronics engineer, a procurement officer, 6–8 PhD/Post.Doc. researchers, and professors supervising the thesis work or offering experience and support. The project manager is a PhD candidate examining the value of MBSE to deliver the CubeSat on time and within schedule. The researchers typically join the project for 2–4 years, and the students for 4 (BSc) or 9 (MSc) months when they write their thesis. The backgrounds of the students vary, but typically they are enrolled in engineering cybernetics, embedded systems, electronic systems, product development, or material science. Some of the students have experience with working in teams, and sometimes multidisciplinary development through previous coursework or volunteer organization. However, not many have experience with product development, which typically has more unknowns than course-organized project work.

The project had its first major milestone in December 2017, the Mission Design Review (MDR). There had been some software development prior to this, mostly focused on algorithm development for processing, without target hardware or system in mind. The overall system maturity timeline is shown in Fig. 3, and a more detailed timeline of the progress in 2020 is shown in Fig. 4. Most of the integration and HIL testing occurred in 2020.

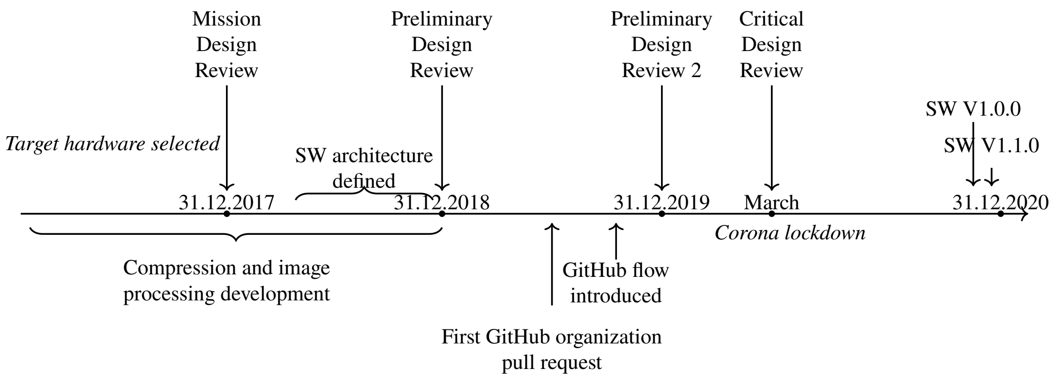


Fig. 3 Overall timeline of in-house developed product maturity, including both hardware and software (SW).

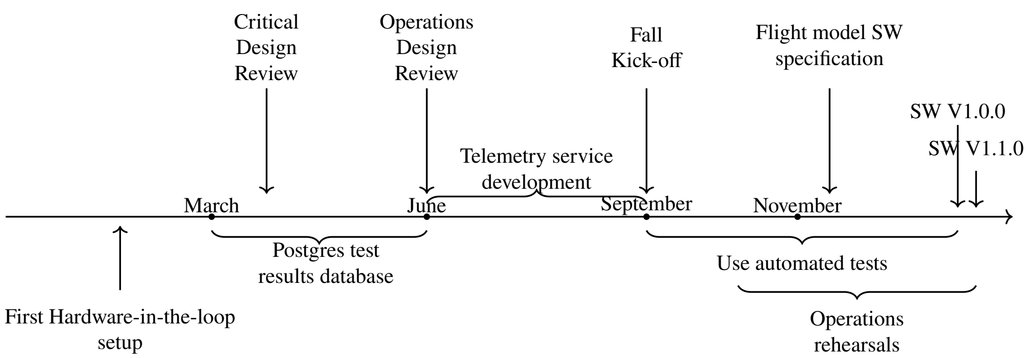


Fig. 4 Timeline of product maturity through 2020. "SW" refers to in-house developed software.

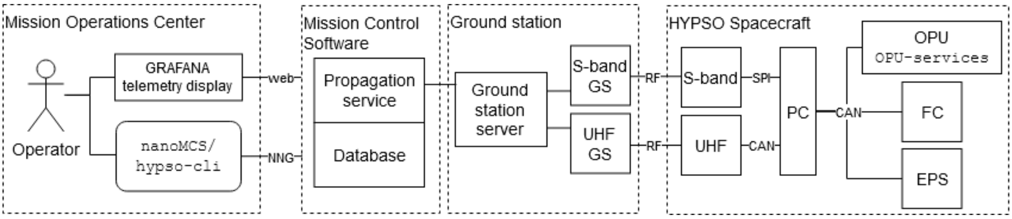


Fig. 5 The software system architecture. OPU = On-Board Processing Unit, FC = Flight Computer, EPS = Electrical Power Subsystem, PC = Payload Controller, CAN = Controlled Area Network, GS = Ground Station, RF = Radio Frequency, NNG = nanomsg Next Generation.

B. Software system architecture

The high-level system architecture is given in Fig. 5, where the flow of signals and data is bi-directional. Some of the items in the software architecture are developed in-house, while others are delivered by suppliers, or interfaced as a service. The architecture was not clear at the beginning of the project, and has been gradually defined throughout the system development lifecycle. The components have also undergone continuous development, as well as updates to the interfaces to a certain degree. The reasons for continuous development and changes are new functionality requirements and new performance requirements, the inherent constraints of the chosen components, as well as the learning and discovery process of developing a CubeSat system for the first time.

Modular software components require that interfaces and software architecture are defined. While the initial software architecture was developed in late 2018, not all interfaces between different components were defined. This meant that a lot of work was required to integrate the in-house developed components. Furthermore, the interface definition to other spacecraft subsystems had not been considered prior to 2018, such that the components also needed adaption to enable integration to the satellite bus. The software-based sub-systems allows for hardware to host the functionality of several subsystems. For the HYPSO spacecraft (Fig. 1), the subsystems "SYS1.3 ADCS Subsystem" and "SYS1.5 OBC Subsystem" are both hosted on the same physical component, the Flight Controller (FC). On the payload, the physical On-board Processing Unit (OPU) hosts the image processing pipeline, the camera control, the payload operating system, and telemetry services for the payload.

In Fig. 5, each partition is composed of tightly integrated physical and software sub-systems; namely, a cyber-physical SoS. The space environment will affect each of the interfaces between the sub-systems and the performance of the spacecraft itself, and the software sub-systems need to adjust (for example pointing the spacecraft towards the sun when the battery levels get low) to ensure functionality and performance. Additionally, this means that to develop hardware components, one needs to consider the software, and when developing software components, one needs to consider hardware limitations, such as data transfer speed limitations, or processing hardware physical layout. Furthermore, the "Mission Control Software" and "Mission Operations Center" were not available until mid-2020, which led to the discovery of new functionality and software adjustments to facilitate operations of the payload. When the spacecraft is operational and commissioned, the operator will only interact with the first box (the telemetry display and the `hypso-cli` (user interface translating commands to packets used for communicating) or `nanoMCS` interface) and the `OPU-services` on the HYPSO spacecraft, under the expectation that the underlying system functions as expected. Despite the many hardware and software systems in between the operator and the spacecraft, they must exchange information correctly and in a timely manner.

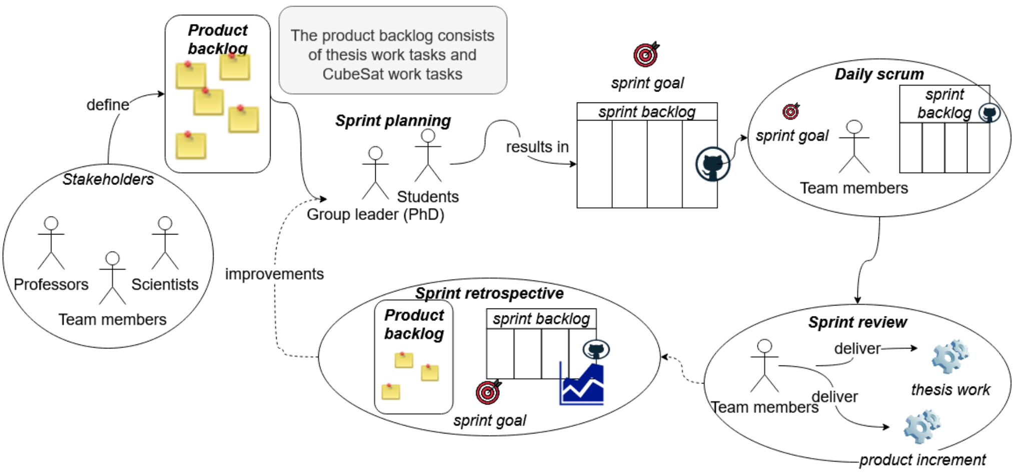


Fig. 6 Tailored Scrum process with a product backlog consisting of both thesis tasks and project work tasks.

C. Tailoring of the agile methodology

The Scrum methodology has been tailored such that the team members deliver either a product increment or a thesis, as shown in Fig. 6. The sprints typically lasted 2 weeks, and there was a daily scrum meeting (a stand-up) in which issues were raised or discussed for clarification, in addition to general keeping-in-touch with each other. The team uses GitHub for managing the code repository and schematics, and providing version control and release management [12]. GitHub is a service that provides users of several different backgrounds and development approaches to work together and at the same time have a coherent overview of the current status of the code base. GitHub has a plugin for managing Scrum with a *kanban* board. Kanban boards, from the Japanese word meaning billboard, are used to visualize and manage workload by providing an overview of work-in-progress, backlogged items, blocked items, done items, and review-in-progress items. A kanban board is based on *pulling* tasks instead of being *pushed*, which enables the students to take control of their own workload. At the same time, the Scrum master (called group leader in Fig. 6) can control which items are included on the board, so the work that gets done is pertinent to the schedule and the product to be delivered.

Planning, workflow, and continuous integration Planning and developing a complex system are not guaranteed to align well with research goals found in academia. Finding synergies and acknowledging what needs to be prioritized can benefit the development of a CubeSat as well as providing a better foundation to build and expand research activities upon. While Scrum traditionally has a goal of delivering a pre-defined Minimum Viable Product (MVP) at the end of a Sprint, this was not the case for HYPSO. In this case study, participants contribute to components ranging from hardware to User Interface (UI). Until the first agreed software release at the end of 2020, as shown in Fig. 4, the sprint backlogs included issues which the team members "wanted to focus on" and had time to work on. There was an agreement between the team members when selecting issues, and there was a continuous focus on working on issues labeled as "bugs" or mission-required functionality (defined by the group leader in conjunction with the project manager) instead of issues categorized as "enhancements" in GitHub. Furthermore, each participant developed modules without defined interfaces between them. This made retaining the value added from different contributions, and especially integration, unnecessarily difficult and time consuming. To mitigate these challenges a common workflow was proposed and became a part of the on-boarding procedure, as well as providing the students with a common repository.

Some of the contributors only participate in the development for as little as one semester, and there are limitations to how complicated the workflow and how complicated the development tools can be. To achieve a convenient workflow, development needs to be coherent and a multitude of development considerations have to be made clear, as well as followed-up to ensure the desired quality of the project and product. Continuous integration (CI), or the practice of integrating contributions from multiple developers into a common software product, is beneficial for collaborative code development [40], and is also promoted in XP practices. A workflow focusing on integration was then proposed, i.e. the GitHub workflow [41]. This workflow states that the main branch shall always be working, and any feature or fix to be included in the code base shall originate from a dedicated branch, i.e. there are no development branches that branch out beyond the main branch. This workflow encourages contributors to frequently merge their code contributions into a central repository for review and testing, as is considered a good practice in software development [41].

D. Verification and validation using Hardware-in-the-loop setups

Verification and validation are important to ensure that the product functions as specified (verification) and meets the needs of an end-user (validation). Collectively, these will be referred to as testing. In the HYPSO project several testing regiments were developed to expand the number of reviewers. The software group leader emphasized that approval of a Pull Request (PR) should be done by reviewers not necessarily involved in the development of the code. In other words, the contributors were required to describe their changes or additions in such a way that "any" software team member could be able to review them. Even though not every team member is able to review every change, this motivates the developer to make code modifications in such a way that they are understandable to “any” person responsible for reviewing said changes. For a change to become part of the master branch, at least one other person has to approve the suggested changes. When the code changes are committed to a separate feature branch of the central repository, it is then built and tested by a team member prior to being accepted as a valid code base addition. If no adverse effects are detected during review, the pending PR is then merged into the master branch. This is the manual process of testing and ensures that specifically the newly added feature or fix is tested independently and sufficiently.

In addition to the manual process, several automatic scripts have been developed to do routine tests of nominal operations of the system. While simplifying the process of testing any proposed changes on the target hardware this also provides a platform for other types of testing. Several installations of the system, laid out as closely as possible with the actual satellite, were set up to be interfaced remotely by any team member, namely the HIL setups. HIL setups can be used for verification of functional requirements [42], and if deployed on target hardware, it can also verify performance requirements. Because university CubeSat projects often have limited funding available, having a full *engineering model* (an exact replica of the system) of the satellite bus and its subsystems is not always feasible. Instead, using a *FlatSat* (a flat satellite) with subsystems provides many of the same functions at a much lower cost. The satellite bus providers often sell FlatSat services at lower fees because the subsystems that constitute the FlatSat can be shared between different customers, or the subsystems can be development models used by the satellite bus providers themselves.

Two HIL setups were developed to facilitate verification and validation activities, and to improve early integration efforts. The HIL setups are shown in Fig. 7, and are called *LidSat* (because the systems are mounted in an ESD-box lid), and *pHIL* (payload HIL). Both setups use target hardware for the software subsystems, and have different purposes. The pHIL setup is mainly for testing payload and its communication interface with the command line interface, while the LidSat is used to test both the payload software and the integration of the payload to the spacecraft. The pHIL is connected to a workstation which is running a Jenkins continuous integration server. To test a branch of the software, the branch is first compiled and initiated on the payload. Then Jenkins runs a set of tests on the target hardware. The outcome of the tests (both whether they pass and their performance) is recorded in a database. The central database allows the developers to see how various branches have performed during the test. The test set includes sending several commands which operators commonly use, and ensuring that the correct results are obtained for different sets of parameters. The LidSat has both the Electrical Power Subsystem (EPS) and payload controller connected via a Controlled Area Network (CAN), with an additional connection to the rest of the spacecraft subsystems on a FlatSat in Vilnius through internet with a CAN-over-internet bridge. These are the main interfaces for the payload, and as such, the FlatSat replicates integration with the spacecraft.

Furthermore, integration testing has been automated by scripting commands to be sent from the operator computer to the payload. Scripts have been developed to aid other hardware team members in testing nominal operations when mechanical changes are made, and these scripts are also used in a test-to-failure scheme where the procedures are

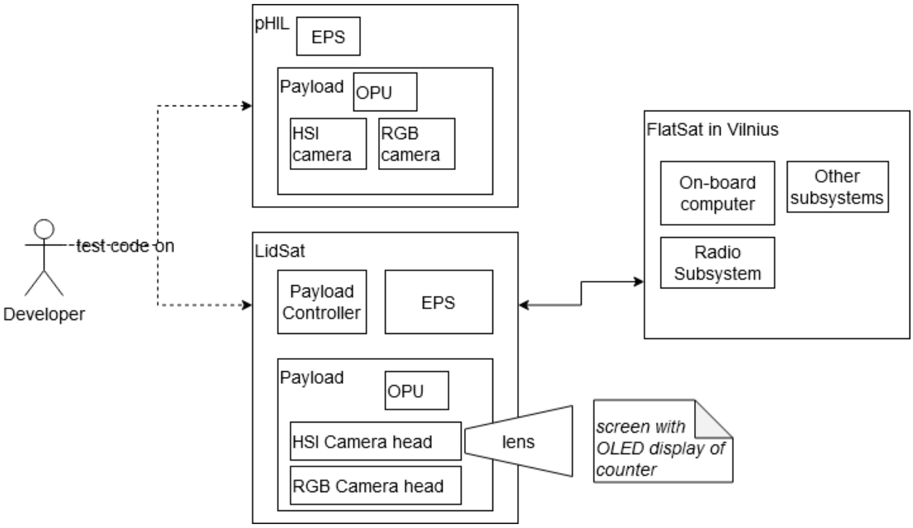


Fig. 7 Hardware-in-the-loop test setups.

repeated a set number of times or until failure. A script testing the potential performance alterations was also used on the system, as well as a test of the subsystem communication and integration. All these tests are run routinely in an effort to uncover unforeseen adverse effects of any proposed code changes.

IV. Experience using digital engineering in an academic project

The product development lifecycle with its DE tools and methods are shown in Fig. 8. Note that specific tools used for analysis are not shown, as they depend on the specific discipline and task the team member is working on. This lifecycle is supported by the GitHub workflow and the Scrum method for daily management of work. There are many improvements that can be made, but the DE strategy presented here is low-cost, and makes use of well-established processes and tools that are readily available. Furthermore, while some training is needed, and there should be an agreement to be consistent, most HYPSO team members agree that the benefits greatly outweigh the cost.

In this section we will discuss which factors influence the approach to DE, evaluate the effectiveness of using agile practices, the educational aspect of the HYPSO project, and also provide some insights gained during the COVID-19 outbreak and how this relates to DE [8].

A. Choice of digital engineering strategy

The choice of DE processes for the HYPSO project team was continuously evaluated, with introduction of new methods and tools as needed. The overall strategy was to adopt and test different DE approaches throughout the project. Typically, the solutions chosen were based on previous knowledge or experience from the team members in other

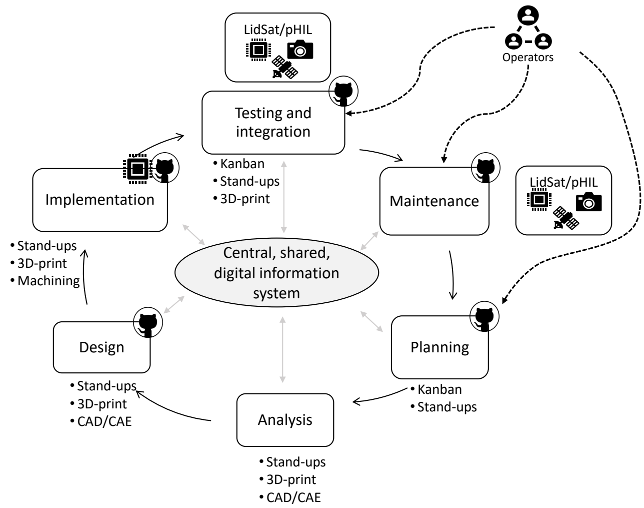


Fig. 8 Product development lifecycle with digital engineering methods and tools.

projects. This previous experience also made training of other team members easier, which is a critical component in the adoption of new methods and tools. From the list in McDermott et al. [36], the factors listed in Tab. 1 were chosen. The factors were selected by reviewing the discussions in the project team that led to the DE approach. No quantitative measures of DE competency before and after introduction of tools were done, however, results from action research have been used as basis for this paper.

Adoption The *DE tools* were based on what would have a high adoption rate, be open-source or free license, and that there would be little resistance from the students. For example, the project team conducted polls to decide on which cloud file repository to use, which communication platform to use, and which video conferencing tool to use. This means choosing tools with good user interface, or tools that have been used in other courses, closely linked to *Workforce knowledge*, to reduce the need for *Training* as there are little *General resources for implementation.* The implementation efforts mainly have to be performed by students or group leaders (PhD candidates). The *DE processes* were selected based on recommendations in literature review [24, 41, 43] and recommendations from other CubeSat teams at informal discussions at conferences such as the International Astronautical Conference or Small Satellite Conference. Considerations were made to find processes that would not require too much *General resources for implementation* and that would quickly *Demonstrate benefits* to the project team, to ensure that the team members were *Willing to use tools*.

Knowledge transfer During the first year of the HYPSO project, challenges with *Information sharing* occurred

Table 1 List of factors influencing digital engineering strategy at HYPSO project. Right-hand side shows the sociotechnical factors, while the left-hand side are more technical.

Digital Engineering Competencies			
Category	**Factor**	**Category**	**Factor**
Quality	Traceability	Knowledge transfer	Better information sharing
	System quality		Better information accessibility
	Reduce defects/errors		Improved collaboration
	Improved system design		Better knowledge capture
	Increased effectiveness		Improved architecture
	Strengthened testing	Adoption	General resources for implementation
Velocity/Agility	Improved consistency		Workforce knowledge
	Reduce time		DE processes
	Increased capacity for reuse		Training
	Early V&V		DE tools
	Easy to make changes		Demonstrating benefits
	Higher level of support for integration		People willing to use tools
		User experience	Improved system understanding
			Reduce effort
			Higher level support for automation
			Better decision-making

frequently, such as missed hardware changes which influenced both software and hardware performance but were not communicated clearly. Furthermore, the complexity of the system necessitates *Better information accessibility* and *Better knowledge capture*, which were two of the main objectives to fulfill for the DE tools and processes chosen. The agile methodology in hardware and software *Improved collaboration* and *Information sharing* both by having the issues documented in GitHub, but also through the common stand-up meetings held daily. In addition to the technical benefits of using the GitHub workflow, having a common workflow could also increase the feeling of team cohesiveness, and shared understanding of how the fragments can work and should work together through for example testing each other's code. The common stand-up meetings enabled a better understanding of how hardware and testing worked for the software developers, and limitations in for example physical interfaces, from the perspective of hardware developers. On the other hand, the hardware developers got a better understanding of how the system would be used operationally, and could align their development and prototyping schedule to accommodate for verification and validation activities.

User experience Because the DE strategy involved stand-up meetings, 3D-printed hardware prototypes, and HIL test setups, team members acquired an *Improved system understanding*. While it is difficult to prove an improvement, discussions during review meetings have been less about clarification and more about design enhancements and future development. The first iteration of the agile methodology used a physical kanban board, which was not adopted well by the team. Introducing a GitHub kanban board *Reduced the effort* needed to separate software code development from the process of managing the development. This is a clear advantage of using DE tools and processes. *Decision-making* has been improved for hardware by employing 3D-printing to prototype and test design alternatives, thus giving more data for making decisions. Automatic unit tests are run on HIL setups before and after software updates are merged to the master branch, providing *higher level support for automation*. However, all unit tests must be developed manually, so there is an effort required there for the developers. The compilation of code generates code documentation in Doxygen automatically. Doxygen can provide information about how functions are related which further helps information accessibility and sharing. Future work could be on enabling more automatic generation of unit tests in parallel with code development.

Velocity and agility The HYPSO project is a part of a long-term strategy for establishing capabilities for developing small satellites for scientific purposes at NTNU [39]. There is thus a need for the development strategy to have a *capacity for reuse* so that the different subsystems can be used across a variety of platforms with some changes, and reused in new satellites. Introducing the different HIL setups have increased the capabilities for *Early V&V*, which has *Reduced time* required to discover bugs. In addition, the increased employment of 3D-printing technology (also a digital technology) in prototyping and the development of Ground Support Equipment (GSE) has reduced the time for hardware development through increased *Early V&V*. Having 3D-printing technology in-house in the lab has made it easier for the team to try out new designs or satellite physical architectures. Furthermore, there is a *Higher level of support for integration* when combining 3D-printed prototypes of hardware, mature HIL setups and test software which can emulate physical conditions such as lost packets on the radio communication link. The GitHub workflow process introduced an *Improved consistency*, together with other standards. The shared repositories enabled students to see how others write code and test, improving consistency across the whole codebase, as well as functioning as a resource for reuse in other platforms or future satellites.

Quality The goal of introducing HIL setups and the GitHub workflow was to *Strengthen testing* and thus *Reduce defects and errors*. However, prior to the introduction of the HIL setups, the Github flow also helped with increased testing and integration into master branch from mid-2019. There were no measures of effectiveness prior to the introduction of DE measures, and the discussion regarding effectiveness is given in Section IV.B. While not considered explicitly when choosing GitHub, the issue tracking and discussion has enabled better *Traceability* of design choices. For example, if a bug or unwanted behavior of code during testing resurfaces, it is possible to search for keywords in GitHub and find similar bugs and investigate if similar solutions can be used to mitigate the unwanted behavior. This can *Reduce the time* spent bug fixing for new developers who were not a part of the project at the time of the original bug. An added benefit from incorporating the design into DE tools such as GitHub, was that it required a conscious decision and discussions regarding *architecture* and *system design* (related to both Knowledge transfer and quality), and there have been three instances of refactoring of code systematically to improve the maintainability and modularity of the codebase.

B. Effectiveness of using agile digital engineering: software and hardware

1. Tailoring of Scrum

The Scrum process was tailored to include issues related to thesis work as well as product development tasks, as shown in Fig. 6. The stand-ups have included both the hardware and the software team, and people could join either physically or with their phone or computer. Most team members have reported that stand-ups have increased their understanding of the system and sharing of information. Some students have reported that the stand-ups increased in relevance as they were working on integration of subsystems, but not so much when they were developing the prototype modules. Another tailoring that was done was to agree on which issues would be performed and ensure that each student had something to work on. This was needed to accommodate thesis work. Unlike traditional Scrum processes, such as the one described by Garzaniti et al. [37], the team did not agree on the functionality for each MVP to deliver at the end of each sprint. In hindsight, a better defined MVP might have improved the results by having a shared goal for each sprint, which can contribute to team cohesiveness and commitment.

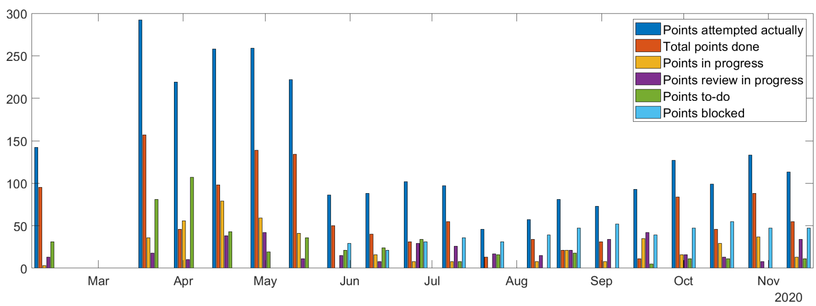


Fig. 9 Full SW Sprint

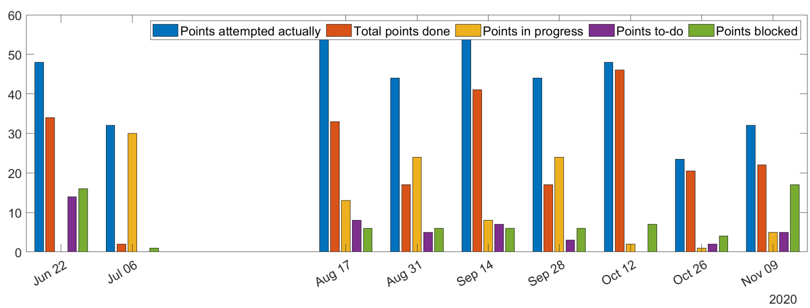


Fig. 10 HW sprint in barplot. There was a break during the summer holiday.

2. Scrum performance

Software The first sprint using GitHub kanban was held in early 2019, and apart from the first sprint, all sprints were two weeks long. The sprints started long after the software development began, and the team had a good enough overview of functionality. The first couple of sprints had a high number of attempted points, with a high "miss-factor" of points not done (February to June). This can be attributed to the learning process and is not uncommon for new Scrum teams. Team members mentioned that it was challenging to figure out how to score their tasks. The Scrum leader can support this process by guiding the students, for example, by referring to previous work they have done and how long it took them to complete. An ongoing challenge has been to have enough reviewers to reduce the amount of points in the "Review-in-progress" column. Since the workflow requires that someone else reviews the code, there needs to be at least one other person with similar knowledge and capabilities to be able to review the code. This may not always be available when the students' priorities are changing to consider coursework and such.

In April 2019, it was decided that the software team leader would be the Scrum leader moving forward, and also run the sprint meeting. Furthermore, that sprint reviews should include an aspect of code demonstration or a more rigorous documentation of how an issue was closed. The team has also discussed how to agree on a "definition-of-done". This definition has not been finalized yet, but there is agreement that it should be related to the type of issue being solved. For example, issues related to theses can be draft sections or chapters, and code issues could be a bugfix, a functioning module or function that has resulted in a PR.

Hardware The hardware team started using the agile framework and sprint methodology at the end of Q2 2020. The payload design had reached a high level of maturity by then, and most of the parts and suppliers chosen. All satellite bus components had been procured. The work that remained was focused on verification and validation activities, and coordination with external test facilities and the in-house mechanical and optical labs. In addition, planning began for the updates of design for HYPSO-2, the next CubeSat to be developed. As shown in Fig. 10, there is a break during the summer holidays. The performance has varied over the nine two-week sprints that have been so far. Many Scrum teams take a while to learn how to estimate points to issues, and to estimate how much work can be done in one sprint. Towards the end of the semester, the total points done matched the points attempted better. This could be because the team became more accustomed to the Scrum workflow, or because the deadline for delivery of the flight model was getting closer and people felt committed to this milestone. The blocked issues were typically due to external factors, such as lack of access to testing or machining facilities, similar to the findings in [37]. There have been continuous redesign and rework activities. The stand-ups helped in coordinating the activities between designers and the group leaders organizing the support facilities. Some hardware team members stated that using Scrum helped them prioritize tasks and not get "distracted" during the two-week period.

However, the greatest issues were related to attendance and commitments to sprints. It was challenging for the group leader to motivate the students when there were too few collaborative tasks. We found that a two-week duration of sprints were suitable for the team because the students were available to deliver increments in that time period. Longer sprints could make it harder to motivate the students, and shorter duration would make it difficult to deliver increments [37]. The motivation could be improved by introducing stricter MVPs or by spending more time planning the work up-front. The MVPs could for example specify new features to be included on the hardware prototypes, iterated simulation results, increased performance or lower manufacturing cost. The MVPs could also be tangible, for example, 3D-printed prototypes and parts that can be validated by other team members, or simulated assembly and incremental tests.

Lessons learned The team experienced challenges with commitment and attendance at stand-up meetings, especially with team members who started during the COVID-19 lockdown (fall of 2020). There were fewer on-boarding and team building activities than previous years, and little or no chance of face-to-face meetings. Some students used the Kanban board to organize their own work, but did not join many of the stand-ups. Based on this experience, we see that it is not sufficient to have good workflows and tools alone, but that the social aspects matter as well. The team members need to be a part of the culture, and people need to feel that they are a part of the team, which is consistent with findings of Garzaniti et al. [37] and Masood et al. [31]. The HYPSO team combined the sprint planning and review meeting to reduce time spent in meetings [31], and adjusted the sprint scoring and length to accommodate the overall school schedules and workload [31, 37].

3. Integration and verification and validation

The HIL setups have been instrumental in easing integration between different systems, both for software development, operations development, and for hardware development. For software development, the HIL setups facilitate not only verification of software changes before merging with the master branch, but also verification that the changes work with the satellite bus via local engineering model versions of subsystems or the FlatSat. There have been HYPSO-initiated interface changes, and NanoAvionics (the satellite bus provider) initiated interface changes. These interface changes have been to improve performance or add functionality. By having a HIL FlatSat-setup with physically distributed subsystems, engineers in Vilnius could update the modules remotely and work concurrently with HYPSO project members. Challenges with the HIL setups included finding people to work on setting them up and developing required functionality, such as automatic tests, and maintaining them. It was also challenging to find sufficiently interesting thesis topics for working with HIL and testing activities.

The HYPSO project team can choose which subsystems they need to locally with the payload (as shown in Fig. 7), and which subsystems that can be located at the supplier premises. The subsystems located at supplier premises can easily receive hardware upgrades without the need for shipping modules back and forth. Additionally, the distributed system still allows the supplier to log in to subsystems located in the university to perform software upgrades, configuration changes or other fixes.

For operations development, the HYPSO operations' developers have been able to perform rehearsals to validate that the software functions and performs as expected. This has been enabled by allowing the operator to connect to the HIL LidSat setup using the *hypso-cli* user interface (as shown in Fig. 5). Experiences from the operators were critical for preparing the first official software release for deployment on the flight model.

4. COVID-19

The COVID-19 pandemic caused the university to lock down on March 12th 2020. Luckily for the team, the HIL setups had been implemented in end of February, which allowed for remote access and testing of software on target hardware. In addition, the regular stand-up meetings had begun the year before, and only required a shift to full virtual meeting. The stand-ups were a bit longer than they had been previously, because more people joined regularly and there was a need to move some of the informal discussions that usually take place in the physical lab to the stand-ups. Team members also said they appreciated the stand-up meetings because it was a forum for social interaction. The issue tracking on GitHub for software helped to follow-up the work and monitor the progress of the project, and was not affected by the lockdown. There was an increase in commits to the main software repositories around the time of the lockdown, and the high frequency persisted until the end of semester, as shown in Fig. 11.

However, no hardware integration and testing could be performed during the lockdown, since the team members were not allowed on campus or to travel to external test facilities. This created a severe schedule delay to the project. The hardware team spent time preparing design documentation and refining test plans until the lockdown restrictions eased.

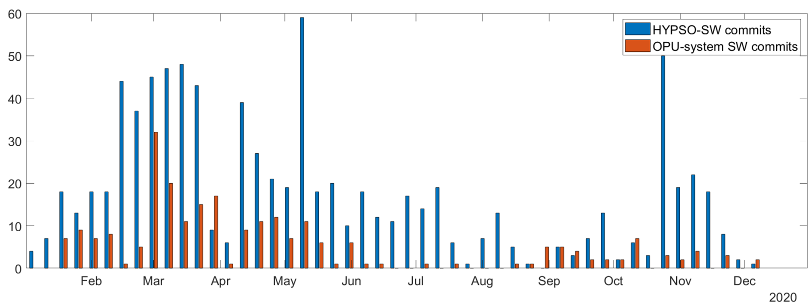


Fig. 11 The two main software repositories commit frequencies.

C. Educational aspects

In the context of digital engineering, the HYPSO project organization described in this paper have many similarities with the projects described in Berthoud et al. [12]. The university CubeSat project format is an inherently interdisciplinary project which prepares students for future work, even if it may be in different industries. Additionally, the use of HIL setups, a strict GitHub development flow, and agile practices in software and hardware development provide the students with a larger skill set for future employers. The students gain practical experience with using digital engineering methods and tools, while still delivering the required coursework and thesis work. While these skills may be gained through capstone courses as well, having an active "customer" with strict deadlines and objectives in addition to educating students, can motivate teams to work even harder with delivering results [32]. The customer for the HYPSO project was the group of scientists who needed the data from the CubeSat, and the deadline was set by the commercial launch date. However, managing CubeSat projects with agile practices requires coordination and training, and should not be underestimated [30–32].

Although we have not done a systematic study of the transferability of skills learned during the HYPSO project, one student mentioned that:

> I have noticed that in my job, where they use Scrum with Kanban on a digital platform, I at once felt at home and prepared for how to do my work. And I also felt that I could contribute fast. The meeting structure and documentation (templates, as-built documents, internal and external design reviews) were similar to how we did it in the HYPSO project, which made it easier for me to see the value of what I had learned and realized the relevance of the HYPSO practices. (...) I felt I was prepared to start a job because I know how the workday is structured and how to organize my work.

Some of the graduated students have joined the team as PhD candidates and taken on leadership roles. The rest of the graduated students have joined companies in various industries, and some still join HYPSO design reviews or contribute to the code repository.

D. Future satellite development

The HYPSO team has started the development of their second CubeSat that will have an upgraded version of the hyperspectral payload, increased processing capabilities, and a Software Defined Radio (SDR) [44]. Based on the experiences from HYPSO-1, the team plans to continue the agile work methodology for both hardware and software, and increase the importance of team building and team cohesiveness. They are also considering introducing MVPs and a clearer "definition of done" [31, 45], which could increase the sprint performance.

The team has introduced a cloud-based digital tool for managing requirements, system budgets, analysis, verification planning, and project planning. Previously, this effort was managed through the systems engineer, but now, the team can collaborate real-time from different sites on the same set of requirements. These updates also feed automatically into system budgets and the product breakdown structure. The team members can create discussions and flag components or requirements, and assign tasks to each other. This is a part of the "Central, shared, digital information system" shown in Fig. 8.

V. Conclusion

Digital engineering is needed for managing the development of complex systems. This requires a conscious effort throughout the organization, and the strategy must be tailored to the specific needs and constraints. There is also a need for engineers who are trained to use digital engineering approaches in their work, in all lifecycle phases of a project. Academic CubeSat projects provide an arena to training future engineers by collaborating in interdisciplinary system development. The students gain both technical and non-technical professional skills. For academic CubeSat projects, the needs for a digital engineering strategy are often similar to the industrial setting, but the context and constraints are quite different.

In this paper we have described the case of an academic CubeSat project in Norway, where they are developing a scientific 6U satellite and ground segment. Because of the challenges with knowledge sharing, unclear decision-making, lack of coordinated planning, and poor code quality and documentation, the project organization introduced some measures that includes digital engineering tools and methods. We have outlined the project development lifecycle, and highlighted how agile practices supported by a digital kanban, a GitHub workflow, and HIL setups have been essential in managing the development of the complex CubeSat. In addition, we have discussed in which ways the digital engineering strategy chosen contributed to verification and validation activities, integration of systems, knowledge sharing, and how the tools and methods supported development even during the COVID-19 lockdown. However, the tools and processes alone are not sufficient for adoption of the DE work environment. People need to be encouraged to use them, and social aspects such as team cohesiveness and commitment are important. Throughout this process the project manager has used a participatory approach in which all team members could influence the practices and processes.

The digital engineering strategy adopted by the HYPSO team is a low-cost, low-effort approach using readily available tools and methods. Some of the methods, such as agile practices and software repositories, have been used in other CubeSat projects. There are valuable lessons to be learned between different academic teams and between industry and academia on how to best approach and implement digital engineering in the organization. Future work will look at including more MBSE tools and incorporating them with the product lifecycle proposed, to increase the common understanding of the system and support knowledge management. Lastly, to combat the hurdles that using target hardware for testing can cause, it is common to simulate the hardware responses. The caveat will always be that the addition of mocking software as well as the addition of unit tests will be prone to the same coding mistakes as any type of software development. The additional overhead of producing and maintaining a mocking library can take away resources from code development that would otherwise provide the needed functionality or enhance it. The addition of unit tests should be added when possible, and could help uncover undesirable side-effects of any proposed changes to the code base.

Future studies could look at: (1) how the graduated studies have experienced transferability of skills and practices gained during the HYPSO project; (2) how other university projects use DE and how the students experience it there; (3) opportunities for cooperation between the CubeSat project and the wider university context, for example by introducing aspects with DE as a part of the student curriculum to prepare for joining cross-disciplinary projects.

Funding Sources

This work is supported by the Norwegian Research Council (grant no. 270959) and the Centre of Autonomous Marine Operations and Systems (NTNU AMOS, grant no. 223254). It has been approved by the Norwegian Centre for Research Data (project no. 560218).

Acknowledgments

The authors would like to acknowledge all the team members at the NTNU SmallSatLab for their participation in the CubeSat project. Authors also thank all the professors and technical support staff at the university departments for their contributions with theoretical insights and practical solutions to developing a CubeSat. E.H-L. would like to thank Vitech Corporation for making the CORE/GENESYS software available for this work.

References

[1] Törngren, M., and Sellgren, U., "Complexity challenges in development of cyber-physical systems," *Principles of Modeling. Lecture Notes in Computer Science*, edited by M. S. Martin Lohstroh, Patricia Derler, Springer, Switzerland, 2018, pp. 478–503. https://doi.org/10.1007/978-3-319-95246-8_27.

[2] Bone, M. A., Blackburn, M. R., Rhodes, D. H., Cohen, D. N., and Guerrero, J. A., "Transforming systems engineering through digital engineering," *The Journal of Defense Modeling and Simulation*, Vol. 16, No. 4, 2019, pp. 339–355. https://doi.org/10.1177/2F1548512917751873.

[3] SEBoK, "Model-Based Systems Engineering (MBSE) (glossary) — SEBoK,," , 2020. URL: "https://www.sebokwiki.org/w/index.php?title=Model-Based_Systems_Engineering_(MBSE)_(glossary)&oldid=59725", Accessed: 2021-02-15.

[4] Aigner, A., and Khelil, A., "Assessment of Model-based Methodologies to Architect Cyber-Physical Systems," *International Conference on Omni-Layer Intelligent Systems*, 2019, pp. 146–151. https://doi.org/10.1145/3312614.3313779.

[5] Beck, K., "Embracing change with extreme programming," *Computer*, Vol. 32, No. 10, 1999, pp. 70–77. https://doi.org/10.1109/2.796139.

[6] Könnöla, K., Suomi, S., Mäkilä, T., and Lehtonen, T., "Can embedded space system development benefit from agile practices?" *EURASIP Journal on Embedded Systems*, Vol. 1, 2017, pp. 1–16. https://doi.org/10.1186/s13639-016-0040-z.

[7] Grande, J., Birkeland, R., Lindem, T., Schlanbusch, R., Houge, T., Mathisen, S. V., and Dahle, K., “Educational Benefits and Challenges for the Norwegian Student Satellite Program,” *Proceedings of the 64th International Astronautical Congress*, 2013.

[8] Honoré-Livermore, E., and Birkeland, R., “Managing Product Development and Integration of a University CubeSat in a Locked down World,” *IEEE Aerospace Conference*, Virtual, USA, 2021.

[9] Puig-Suari, J., Turner, C., and Ahlgren, W., “Development of the standard CubeSat deployer and a CubeSat class PicoSatellite,” *2001 IEEE aerospace conference proceedings (Cat. No. 01TH8542)*, Vol. 1, IEEE, 2001, pp. 1–347. https://doi.org/10.1109/AERO.2001.931726.

[10] Sweeting, M. N., “Modern Small Satellites-Changing the Economics of Space,” *Proceedings of the IEEE*, Vol. 106, No. 3, 2018, pp. 343–361. https://doi.org/10.1109/JPROC.2018.2806218.

[11] Kulu, E., “NanoSats database,” online, 2020. URL: https://www.nanosats.eu, Accessed 2020-12-11.

[12] Berthoud, L., Swartout, M., Cutler, J., Klumpar, D., Larsen, J. A., and Nielsen, J. D., “University CubeSat Project Management for Success,” *Proceedings of the Conference on Small Satellites*, 2019. https://digitalcommons.usu.edu/smallsat/2019/all2019/63/.

[13] NASA, “State-of-the-ArtSmall Spacecraft Technology,” Tech. Rep. NASA/TP—2020–5008734, NASA, 2020. URL: https://www.nasa.gov/sites/default/files/atoms/files/soa2020_final3.pdf.

[14] Dawson, M., Burrell, D., Rahim, E., and Brewster, S., “Integrating Software Assurance into the Software Development Life Cycle (SDLC),” *Journal of Information Systems Technology and Planning*, Vol. 3, 2010, pp. 49–53.

[15] Jayaram, S., and Swartwout, M., “Significance of Student-Built Spacecraft Design Programs: Its Impact on Spacecraft Engineering Education over the Last Ten Years,” *2011 ASEE Annual Conference & Exposition*, 2011. https://doi.org/10.18260/1-2--18345, URL https://peer.asee.org/18345.

[16] Howell, E., “CubeSats: Tiny Payloads, Huge Benefits for Space Research,” , 2018. https://www.space.com/34324-cubesats.html, Accessed 2020-12-10.

[17] Straub, J., and Whalen, D., “Evaluation of the Educational Impact of Participation Time in a Small Spacecraft Development Program,” *education sciences*, Vol. 4, No. 141-154, 2014. https://doi.org/10.3390/educsci4010141.

[18] Grande, J., Birkeland, R., Gjersvik, A., Mathisen, S. V., and Stausland, C., “Norwegian student satellite program – lessons learned,” *Proceedings of The 68th International Astronautical Congress*, 2017.

[19] NASA, “ELaNa - Educational Launch of Nanosatellites,” online, 2020. URL https://www.nasa.gov/mission_pages/smallsats/elana/index.html, accessed 2020-12-11.

[20] ESA, “CubeSats - Fly Your Satellite!” online, 2020. URL https://www.esa.int/Education/CubeSats_-_Fly_Your_Satellite, accessed 2020-12-11.

[21] Paluch, S., Antons, D., Brettel, M., Hopp, C., Salge, T.-O., Piller, F., and Wentzel, D., "Stage-gate and agile development in the digital age: Promises, perils, and boundary conditions," *Journal of Business Research*, Vol. 110, 2020, pp. 495–501. https://doi.org/10.1016/j.jbusres.2019.01.063.

[22] Cooper, R. G., "The drivers of success in new-product development," *Industrial Marketing Management*, Vol. 76, 2019, pp. 36–47. https://doi.org/10.1016/j.indmarman.2018.07.005.

[23] Merriam-Webster Dictionary, *Agile*, 2020 (accessed October 13, 2020). https://www.merriam-webster.com/dictionary/agile.

[24] Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D., *Manifesto for Agile Software Development*, 2001. URL: https://agilemanifesto.org/, Accessed October 12, 2020.

[25] Wilson, G., Aruliah, D. A., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K. D., Mitchell, I. M., Plumbley, M. D., Waugh, B., White, E. P., and Wilson, P., "Best Practices for Scientific Computing," *PLOS Biology*, Vol. 12, No. 1, 2014, p. e1001745. https://doi.org/10.1371/journal.pbio.1001745.

[26] Arvanitou, E.-M., Ampatzoglou, A., Chatzigeorgiou, A., and Carver, J. C., "Software engineering practices for scientific software development: A systematic mapping study," *Journal of Systems and Software*, Vol. 172, 2021, p. 110848. https://doi.org/10.1016/j.jss.2020.110848.

[27] Sletholt, M. T., Hannay, J., Pfahl, D., Benestad, H. C., and Langtangen, H. P., "A literature review of agile practices and their effects in scientific software development," *Proceedings of the 4th International Workshop on Software Engineering for Computational Science and Engineering*, Association for Computing Machinery, Waikiki, Honolulu, HI, USA, 2011, pp. 1—-9. https://doi.org/10.1145/1985782.1985784.

[28] Storer, T., "Bridging the Chasm: A Survey of Software Engineering Practice in Scientific Programming," *ACM Comput. Surv.*, Vol. 50, No. 4, 2017. Article 47, https://doi.org/10.1145/3084225.

[29] Rother, M., and Rosenthal, M., "An Approach to Becoming Agile in a Dynamic World. Helping employees develop scientific thinking empowers them to solve problems and make decisions," *Association for Manufacturing Excellence: TARGET*, Vol. 34, 2018.

[30] Sharp, J. H., and Lang, G., "Agile in teaching and learning: Conceptual framework and research agenda," *Journal of Information Systems Education*, Vol. 29, No. 2, 2018, pp. 45–52. http://jise.org/Volume29/n2/JISEv29n2p45.html.

[31] Masood, Z., Hoda, R., and Blincoe, K., "Adapting Agile Practices in University Contexts," *Journal of Systems and Software*, Vol. 144, No. Special Issue on Software Engineering Education and Training, 2018, pp. 501–510. https://doi.org/10.1016/j.jss.2018.07.011.

[32] Lundqvist, K., Ahmed, A., Fridman, D., and Bernard, J., "Interdisciplinary Agile Teaching," *2019 IEEE Frontiers in Education Conference (FIE)*, 2019, pp. 1–8. https://doi.org/10.1109/FIE43999.2019.9028544.

[33] Shah, M., Grebennikov, L., and Nair, C. S., "A decade of study on employer feedback on the quality of university graduates," Vol. 23, 2015, pp. 262–278. https://doi.org/10.1108/QAE-04-2014-0018.

[34] Støren, L. A., Reiling, R. B., Skjelbred, S.-E., Ulvestad, M. E. S., Carlsten, T. C., and Olsen, D. S., "Utdanning for arbeidslivet: Arbeidsgivers forventninger til og erfaringer med nyutdannede fra universiteter, høgskoler og fagskoler/Education for employment: The employers expectations for and experiences with newly graduates from universities, colleges and vocational schools," Tech. rep., Nordic Institute for Studies in Innovation, Research and Education, 2019.

[35] Turner, R., "Toward Agile Systems Engineering Processes," *The Journal of Defence Software Engineering*, Vol. 20, 2007, pp. 11–15.

[36] McDermott, T., Van Aken, E., Hutchison, N., Blackburn, M., Clifford, M., Chean, N., Salado, A., and Henderson, K., "Summary Report Task Order WRT-1001: Digital Engineering Metrics," Tech. rep., Stevens Institute of Technology, 2020.

[37] Garzaniti, N., Briatore, S., Fortin, C., and Golkar, A., "Effectiveness of the Scrum Methodology for Agile Development of Space Hardware," *2019 IEEE Aerospace Conference*, 2019, pp. 1–8. https://doi.org/10.1109/AERO.2019.8741892.

[38] Huang, P. M., Darrin, A. G., and Knuth, A. A., "Agile hardware and software system engineering for innovation," *2012 IEEE Aerospace Conference*, 2012, pp. 1–10. https://doi.org/10.1109/AERO.2012.6187425.

[39] Grøtte, M. E., Birkeland, R., Honoré-Livermore, E., Bakken, S., Garrett, J. L., Prentice, E. F., Sigernes, F., Orlandic, M., Gravdahl, J. T., and Johansen, T. A., "Ocean Color Hyperspectral Remote Sensing with High Resolution and Low Latency - the HYPSO-1 CubeSat Mission," *IEEE Transactions on Geoscience and Remote Sensing*, 2021. In press.

[40] Fowler, M., and Foemmel, M., "Continuous integration," , 2006. https://moodle2019-20.ua.es/moodle/pluginfile.php/2228/mod_resource/content/2/martin-fowler-continuous-integration.pdf.

[41] Chacon, S., and Straub, B., *Pro git*, Springer Nature, 2014. https://doi.org/10.1007/978-1-4842-0076-6.

[42] Corpino, S., and Stesina, F., "Verification of a CubeSat via hardware-in-the-loop simulation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 50, No. 4, 2014, pp. 2807–2818. https://doi.org/10.1109/TAES.2014.130370.

[43] Waswa, P. M. B., and Redkar, S., "A survey of space mission architecture and system actualisation methodologies," *International Journal of Aerospace Engineering*, Vol. 4, No. 3, 2017, p. 38. https://doi.org/10.1504/IJSPACESE.2017.085674.

[44] Quintana-Diaz, G., Birkeland, R., Ekman, T., and Honoré-Livermore, E., "An SDR Mission Measuring UHF Signal Propagation and Interference between Small Satellites in LEO and Arctic Sensors," *Small Satellite Conference*, Logan, UT, 2019.

[45] Garzaniti, N., and Golkar, A., "Performance Assessment of Agile Hardware Co-development Process," *2020 IEEE International Symposium on Systems Engineering (ISSE)*, 2020, pp. 1–6. https://doi.org/10.1109/ISSE49799.2020.9272209.

Paper F

Honoré-Livermore, Evelyn; Lyells, Ron; Garrett, Joseph L.; Angier, Robert (Rock); Epps, Bob, "An Agile Systems Engineering Analysis of a University-built CubeSat," IS2021 INCOSE International Symposium on Systems Engineering, 31, pp. 1334–1348 [17]. `https://doi.org/10.1002/j.2334-5837.2021.00904.x`

Here, we provide the results of an agile analysis of a university CubeSat organization, and suggest areas for improvement.

31st Annual INCOSE international symposium
Honolulu, HI, USA
July 17 - 22, 2021

An Agile Systems Engineering Analysis of a University CubeSat Project Organization

Evelyn Honoré-Livermore
Norwegian University of Science and Technology (NTNU)
7491 Trondheim
+47 400 18 398
evelyn.livermore@ntnu.no

Ron Lyells
Honeywell (retired)
+1 505-263-1893
rlyells@aol.com

Joseph L. Garrett
NTNU
7491 Trondheim
joseph.l.garrett@ntnu.no

Rock Angier
IBM (retired)
+1 919-233-8029
rangier@mindspring.com

Bob Epps
Lockheed Martin Corporation (retired)
+1 610-247-0961
repps999@aol.com

Abstract. University CubeSat projects become popular in recent decades, and face challenges that include both technical and sociotechnical aspects. However, these teams often lack the infrastructure and resources for having effective systems engineering or project management which are beneficial for addressing these challenges and developing complex systems, such as satellites. In this paper we present the results of an exploratory case study of a university CubeSat team developing an Earth Observation satellite. The Agile Decision Guidance method was applied to pinpoint parts of the project organization that could benefit from agile methods in three specific areas: customer problem space, solution space, and product development space. The results drew attention to areas such as; stakeholder management, knowledge and information management, and the support environment, that could benefit from an agile approach. We outline some of the plans to move forward and how the team responded to the analysis. We also discuss if the method was appropriate for academic small satellite organizations and adaptations of the method made during the assessment.

Introduction

Even though academic CubeSat projects have grown in popularity in the past decades, but still only have a success rate of 60% (Berthoud et al. 2019). These projects apply a mix of ad hoc practices, streamlined processes, or standard space system practices (Swartout 2019). The low success rate is attributed to multiple factors such as a lack of consistent processes and support functions, inadequate knowledge transfer because of high turnover based on the academic calendar and graduations, a lack of dedicated project management (PM), little time for verification and validation, and schedule overruns (Grande et al. 2017).

The environment of a university CubeSat project is typically characterized by 15-30 volunteer students, a few professors, and a small team of researchers (Cho et al. 2017). They lack resources or support systems for stringent process requirements and quality management, or dedicated support functions (such as procurement or mechanical workshops), as is typical in larger aerospace and defense companies that historically developed satellites.

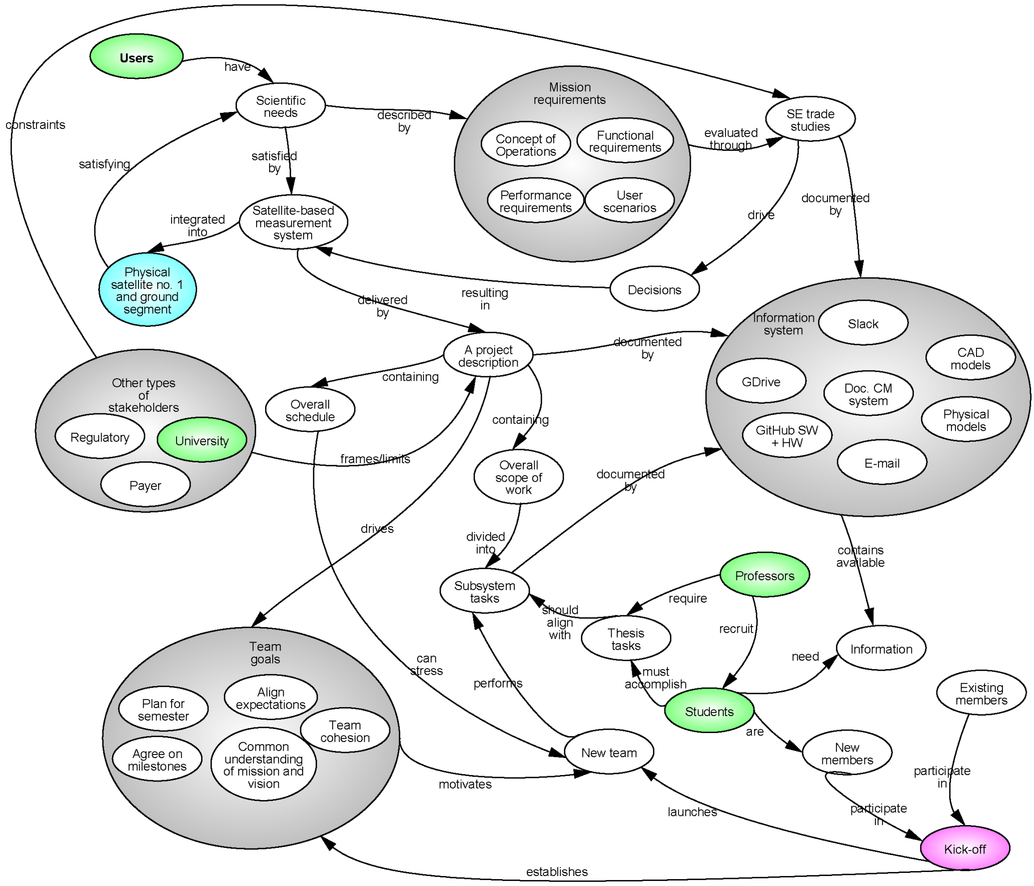


Figure 1. High-level systemigram of CubeSat project organization. The light green boxes are stakeholders, the light blue the final system, and purple a physical meeting.

The Norwegian University of Science and Technology (NTNU) has a goal of building the capabilities needed to rapidly deliver scientific CubeSat missions for maritime surveillance. The first CubeSat mission, started in 2017, had an original schedule of 2 years. However, it was delayed for several reasons, and will be launched in 2021. The team would like to shorten the development time from ideation to launch-ready to 1 year on average. Because of the limited resources available, and the uncertain circumstances in the university context, it was decided to take an agile approach to the introduction of system engineering (SE) processes, instead of, for example introducing the full body of the European Space Agency (ESA) or the National Aeronautics and Space Administration (NASA) standards commonly applied in space projects.

A high-level systemigram in Figure 1 shows some aspects of the project organization and its information system. University students must balance the thesis work and project work – balancing the needs and expectations of their professors with that of the project team. Moreover, there is a high turnover every ~9 months for when the students graduate.

Agile SE “refers to the adaptability and sustainment of adaptability” (Dove, LaBarge 2014, 859), and is concerned with being agile in both the process applied to developing systems – and to the systems themselves. The agile method has perhaps gained the most popularity in software development through “Manifesto for Agile Software Development” and is increasingly applied in the hardware and other domains as well. The INCOSE Agile SE Working Group develop an “Agile Decision Guidance (ADG) Method” (Lyells et al. 2018). This method was applied to the project after reviewing a “Project story”, a narrative describing the development of a university CubeSat system, together with the method’s inventors.

In this paper, we show how the issues described in the “Project story” helped uncover some of the challenges with developing CubeSats in this academic project, such as flow of work, goal alignment, support environment, and knowledge transfer. Then, we discuss how the ADG method was used to study the CubeSat project, and what it indicated about the CubeSat project organization.

Understanding the Problem Space

The OODA-loop, conceived by John Boyd, is the basis of the ADG method (Lyells et al. 2018). The OODA-loop consists of the following four activities: Observe-Orient-Decide-Act. This paper describes the results from the activities in the first three components of the OODA-loop, which is reflected in Figure 2.

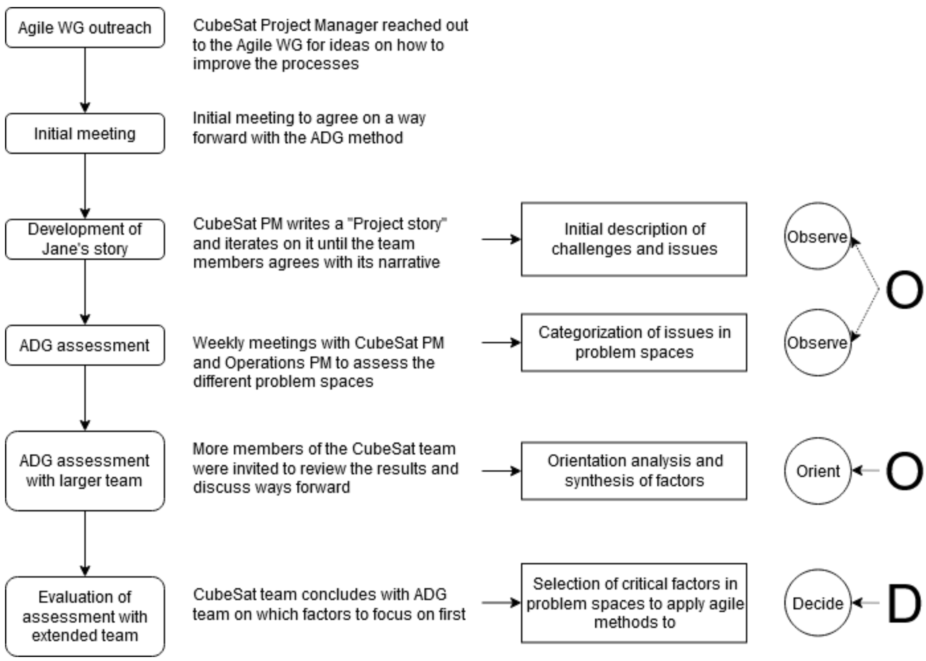


Figure 2. Workflow applied in this project. The phases of the OODA-loop are indicated on the right-hand side.

Observation: Project Story for Academic CubeSat Team

The “Project story”, given in the Appendix, was used to describe the current project from the project manager’s perspective on the culture and behavior of the development. The university had not had any successful CubeSat missions previously, and the faculty members had little or no experience building satellite systems. Furthermore, the university does not have a formal SE program, although some courses are available in the Department of Mechanical and Industrial Engineering. The CubeSat team mostly consisted of electronics and cybernetics students, which are fields typically characterized by reductionist approaches.

The Project story was used to develop team alignment with the problem definition. Team members were asked to read through and then meet for a workshop to improve and change parts so that there was an agreement on the situation described. The Project story revealed problems of development such as flow of work (knowledge transfer, rework time, decision-making, authority, common processes), goal alignment between stakeholders (students for their thesis vs. for project work, professors wanting a fast delivery of CubeSat in parallel with research articles, engineering vs. research goals), and information flow. These system development problems were sociotechnical in nature, and could be linked partially to the failure to adopt a clear SE approach, evident through lack of good requirements handling, or systematic decision-making. This was attributed in part to one person having to simultaneously fulfill the role of researcher, project manager, and systems engineer. During the writing and in-team orientation of the Project story, some participants indicated that multiple information sources caused confusion – and that it was challenging to be the first team to work on a CubeSat in the university, but that they were hopeful that it would become better as the organization matured.

A challenge with the Project story is keeping it up to date, since the organization is not static and there will be new issues appearing as the projects move through their lifecycles. It is also difficult to express the different viewpoints from the team members, for example if they are in different positions or have different academic profiles (student vs. professor vs. Ph.D.). The story was not used actively during the next assessment phase but was available for reference if needed and, it is a good starting point for discussing issues and grounding the problem definition.

Observation: Assessment Questionnaire

The assessment questionnaire is divided into three observation spaces: (1) customer problem space, (2) solution space, and (3) product development space (Lyells et al. 2018). These observation spaces are used to describe the organization and mission context, looking at factors such as stakeholders, requirements, goals, modules, solution sets, development environments and support environments. These factors are all part of the SE context. There is also a fourth component, which assesses the team's agile response capabilities. Each space has a set of questions for characterizing the stability, variety, observability, and predictability of a factor. Thirty-six factors were assessed by answering 111 questions, which were scored between 1-7. An example is shown in Figure 3.

Decision Tool: Stability and Predictability Grid *template v2.0f* 6-May-2020

Project HYPSO | Assessor | updated | Date 7-May-2020

Customer Problem Space - Dynamics

C-CS. Customer Set *Consider the customer or customers sets that direct, acquire, approve, fund and/or use or maintain the target system*

	Variability	Answer	Comments	Score	1	2	3	4	5	6	7	Interpretation
CCS1	How many distinct customer sets are there?	5 groups	funding body Research Board is not customer researchers science board uses satelite data Operators direct development, drawn from masters students Students' customers are professors and PM	5	*1*	*2*	*5*	*10*	*20*	*50*	*100+*	Drives single or multiple solutions, approvers
CCS2	How diverse are the customer sets?	highly varied	all are in knowledge-intensive areas, vary in: goals, stage of life, backgrounds. Some groups representation changes every year	4	Uniform	a few differences	Varied but overlapping	highly varied	Each is unique			One problem space or many
	Predictability											
CCS3	How predictable are customer set changes?	very predictable	We know when people are leaving. Res. Council & professors don't change	1	very predictable	predictable	somewhat predictable	unpredictable	very un-predictable			How well do we know of upcoming changes?
CCS4	How far in advance can customer set changes be known?	6 months (semester)	students are from 3-9 months	5	decade	5 years	3 years	1 year	6 months	3 months	1 month	potential lead time to make changes
	Stability											
CCS5	How often has the customer set changed?	6 months (semester)		5	decade	5 years	3 years	yearly	6 months	3 months	monthly	quantified historic result
	Observation											
RCS1	How often do we recognize upcoming changes to the customer set before-hand?	always		3	always	75%	50%	25%	never	no change has occurred		quantified historic result

Figure 3. Typical question format

The assessment questionnaire was introduced to the CubeSat project manager and the CubeSat operations manager (hereafter called the subjects) in a kick-off meeting. Prior to this meeting, the subjects had acquainted themselves with the method through the original paper by Lyells et al. (2018). The first session was as much assessment as familiarizing the subjects with the questions and format. Two representatives were sufficient to describe the organization, because the subjects both have managerial responsibilities and do engineering as well as other project tasks. There were some discussions between the subjects when answering each question, and the rationale for each answer was recorded in the "Comments" row as shown in Figure 3. In the first session, 12 questions were covered. Each of the remaining 6 sessions covered between 18-25 questions.

Throughout the sessions, the subjects gave feedback on both the ADG method and the questionnaire, either when the wording or questions were unclear, or when it was difficult to give an answer. For example, after three sessions, it was determined that the project was too complex to be described by a single score for each question. The questionnaire was then updated to take in a range of scores, so that different aspects of the project were represented. For example, when discussing the solution space, software modules can be updated regularly, and functional and logical allocation may change – which may change the solution. On the other hand, the physical ground segment does not change much after it has been procured.

Orientation: Agility Factors

After each observation space was assessed, the factors were analyzed to give insight into *Dynamics, Variety and Visibility*. These characteristics helped the team understand how the factors compared in each observation space, and where they could focus efforts on improving:

- **Dynamics:** The space's dynamics show how the project changes over time, the predictability of those changes, and how the team responds. The dynamics analysis influences the types of actions needed. Analysis guidelines are given in Lyells et al. (2018). For example, if a factor has a low stability but high predictability, the response should be rapid and can follow a known template. An example of dynamics analysis is shown in Figure 4.
- **Visibility**: The visibility shows the change rate vs. lead time for implementing changes. It reflects how soon changes can be known and how quickly the team can react to them. If the changes happen more often than the lead time to implement changes required, it becomes challenging to keep up. An example of visibility analysis is shown in Figure 5.
- **Variety**: The variety shows the number of and uniqueness of instances, which affects the mechanisms and capacity required to respond. A team would require a larger amount of effort to manage a high number of instances, and if they are very diverse, each instance may require a tailored effort. Applying a systematic approach to managing each instance and to seeing the differences between each instance could be helpful, as opposed to an ad hoc approach. An example of the variety analysis is shown in Figure 6.

A missed detection factor was calculated based on historic experience data gathered from the questionnaire. The missed detection rate measures the effectiveness of the project's observation system, where a higher value means a less effective observation. For example, the support environment factor had a high dynamic factor and missed detection rate and is coupled to the challenges and constraints that shape the support environment (a university), as shown in Figure 7.

A dynamics factor for the CubeSat team, shown in Figure 8, was calculated based on the frequency of changes and the capacity to anticipate changes, which represents the interplay of these factors from:

$$D_f = \sqrt{Instability^2 + Unpredictability^2}.$$

Moreover, the effort required from the team depends both on the number of the tasks to complete, and the diversity of the tasks. Here, diversity means that the tasks need tailored approaches to be completed. Thus, the scope of impact factor is defined as:

$$I_f = \sqrt{Diversity^2 + (\log_{10}\frac{Quantity}{2})^2}.$$

The missed detection factor, the dynamics factor, and the impact factor were charted in bar charts to enable comparison of factors across the observation spaces and to look for coupling to other factors. A high score represents a greater challenge to the team for all factors.

The factors that were identified as high in dynamics shown in Figure 8 were: *support environment, challenges and constraints, stakeholders,* and all *Solution set factors.* The factors with high scope of impact shown in Figure 9 were: s*upport environment, solution interfaces, stakeholders, module set, customer set, mission environment* and *construction environment.*

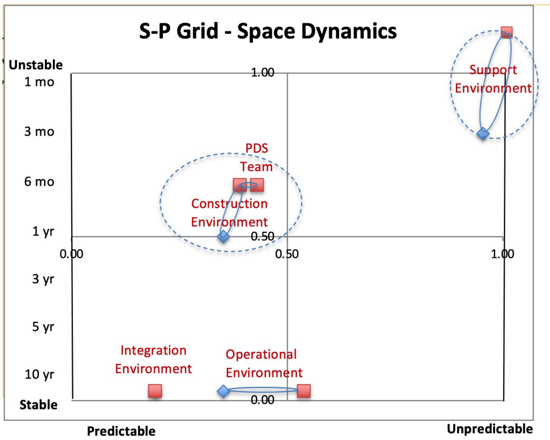


Figure 4. Dynamics view of the Product Development Space.

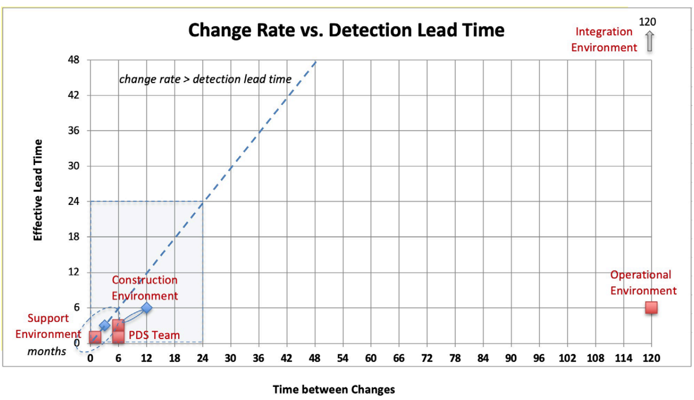


Figure 5. Change rate vs. detection lead time of the Product Development Space.

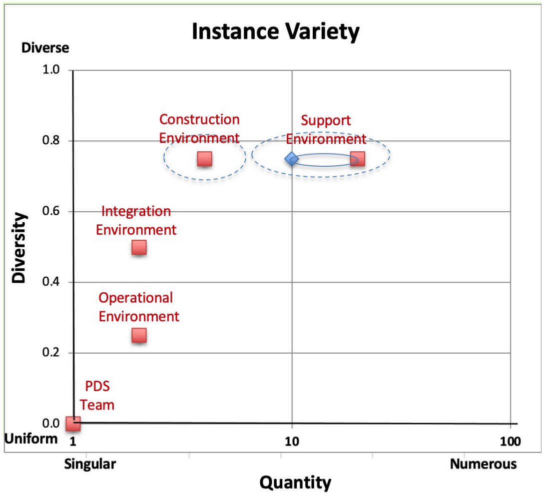


Figure 6. Variety view of the Product Development Space

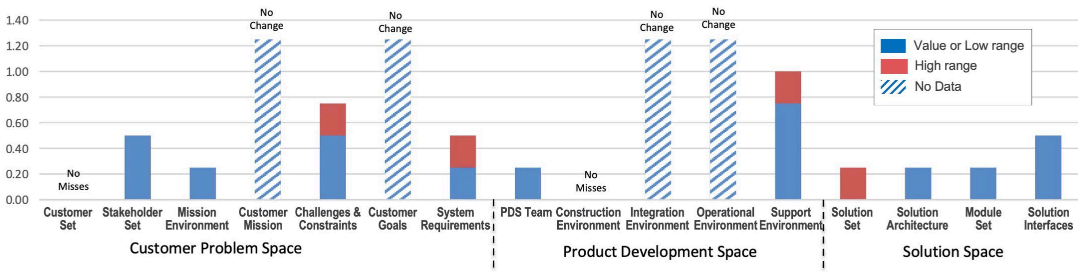


Figure 7. Missed detection of factors for CubeSat team based on historic data.

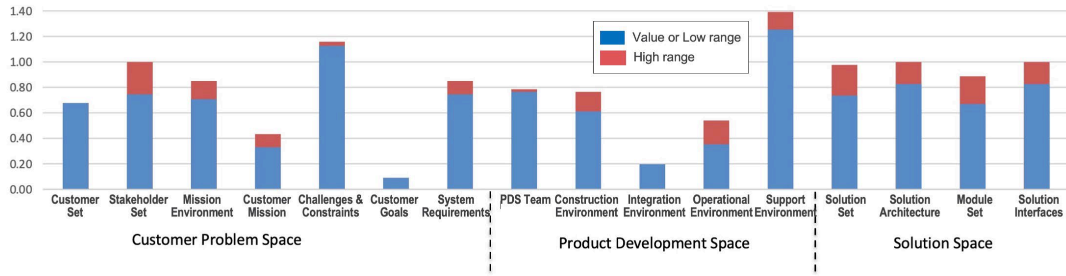


Figure 8. Dynamics of factors for CubeSat team.

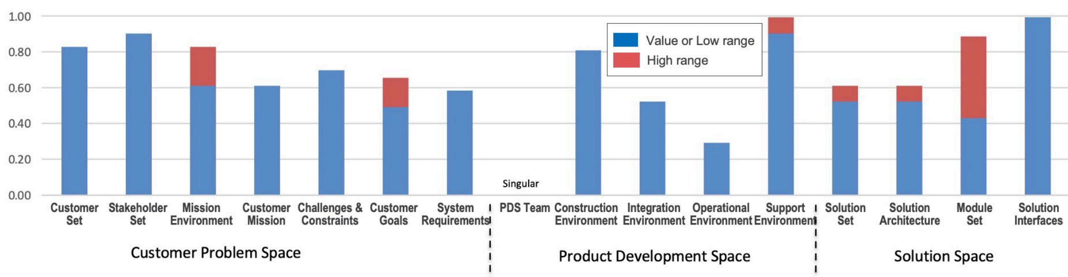


Figure 9. Scope of impact of factors for CubeSat team.

Customer Problem Space

Within the customer problem space, the following factors were assessed: *customer set, stakeholders, mission environment, customer mission, challenges and constraints, customer goals, and system requirements.* In this academic project, the set of customers and stakeholders were highly varied yet identifiable. For example, customers include: professors interested in students graduating or funding more research projects, students wanting a good grade and experience preparing them for the workforce, and end-users wanting the satellite data products. The commonality between the customers identified was that they all belong to knowledge-intensive areas. Most of the customers are relatively invariant, except that the group of students will change every year, and each individual may have different goals and needs for the project and mission.

The organization context is the university, which imposes unique challenges and constraints. Students are both the main source of resources, and a limited resource. Moreover, training is often required because they do not always have a complete skill set or experience needed when they join the team. There may be goal misalignments, as shown in the systemigram in Figure 1, such as students wanting to get an easy grade (which has not been the experience of the team so far) and the project needing engineering tasks or assembly tasks that do not contribute directly to thesis writing. The resource availability is limited to the academic year so tasks that are not completed before the students graduate must be postponed until the new team members arrive. If the new team members do not have the knowledge needed to perform those tasks, further delays can accrue.

In addition, the project manager does not have incentives that can be used for external motivation of students but relies on building team spirit and ownership of project to create intrinsic motivation factors. Most of the reward systems in universities are built on rewarding the individual, through exams or coursework throughout the semester. When students then join a project team, but their thesis work still relies on individual assessment, there is no reward structure in place to encourage teamwork.

In the customer problem space, most of the factors have a lead time that equals the change rate. However, if we look at the scope of impact of these factors in Figure 9, all the factors have a high score. This means that even though the team can change and adapt, it still costs them much in terms of effort.

Figure 7 shows that the project has historically missed changes in *stakeholders, mission environment, mission challenges and constraints* and *system requirements*. The background for a "high miss" in mission challenges and constraints was the delayed development of an integrated testing setup because there were not enough resources. Once the team started integrating subsystems, missing performance and functionality was discovered, leading to an adaptation of the system requirements. Some of the missed detections of system requirements may be attributed to the lack of experience of the whole regulatory framework since it was the first time the team has built a CubeSat, while other misses are due to ambiguity of the ownership of requirements. These stem from having an initial SE approach that was too complicated to be implemented with the size of the team. Requirements management and verification and validation are key SE activities.

During the assessment, the team successfully made some efforts to improve the requirements management. The requirements were moved from being managed in an excel spreadsheet to a SE-enabled software tool. In the software tool, the requirements had required attributes such as: ID, Name, Description, Rationale, and relationships to Verification Requirements, Verification Type, Parent Requirement, Child Requirement. All Verification Requirements were allocated to specific events (such as environmental testing or analysis deadlines) and assigned to individuals as owners.

Solution Space

The assessment of the solution space looked at *solution set, solution architecture, module set, and solution interface.* The software has been developed with multiple modules, which allows for individual thesis work, as is expected from the university. The downside of this is that the module set has been very unstable, and sometimes the interfaces have not been clear at the time of development. This has created integration issues, such as a malfunctioning codebase or rework on other modules. Additionally, there have been instances when the module interfaces have changed but the changes were not discovered until after the student has left, and residual code was not documented well enough to understand or troubleshoot. The project began without a defined software architecture which could have provided the interface management and understanding of how modules were interdependent. The discussion of solution interfaces started the topic of knowledge management, which was stated in the Project story – how sometimes changes are made but not communicated well or absorbed well by the rest of the team, leading to delays and confusion. Interface and knowledge management, in addition to being technical challenges, are also sociotechnical SE challenges.

An N-Squared chart had been created to show the dependencies between people and their interfaces, and how their work was dependent on others (Honoré-Livermore 2019). However, this was not revisited as often as it perhaps should have been, so it was not reinforced, but could have helped the interface management.

Product Development Space

In the product developmen space, *construction environment, integration environment, operation environment, support environment, and product development team* were assessed. As mentioned earlier, the product development team has a high change rate, but is very predictable since the academic calendar is consistent. The support environment is the most challenging factor for this CubeSat project, reflected both in the Project story and in the questionnaire assessment. The dynamics of the support environment is the highest of all factors, as shown in Figure 8. In Figure 9, the support environment also has a high impact on the project team. For example, a delay in accessing a laboratory during assembly of the hardware will delay the schedule for the whole project, because it is not straight-forward to replace or mimic the capabilities the laboratory provides. During the discussions, it became clear that most of the support environment activities are ad-hoc, such as supplier management, procurement, quality control, workshop access, and laboratory access. Furthermore, that there is not a transparent prioritization process in allocating supporting resources at the university.

On the other hand, the construction and integration environments are fully under control of the project team. They can choose which tools and methods they use to build the system. Although the tools vary between specific tasks due to the complexity of the system, changes to the environments are predictable. During the assessment, it became evident that the team has not worked much with the satellite operations environment because of lack of qualified people. Thus, it was difficult to evaluate some of the questions associated with the operational environment.

Product Development Agile Response Capability

The final characteristics to be assessed were the product development agile response capabilities. The results are shown in Figure 10 and Figure 11, where the factors have been categorized into “Communicate and Orient”, and “Decide, Act, and Observe”. Communication is an underlying capability needed for the team to execute the OODA-loop shown.

Communicate and Orient: The team used a variety of tools to communicate, such as Slack, e-mail, GitHub, in-person meetings and shared online drives. They used a mix of push (GitHub) and pull (online drives) of information, and there was a consensus that the scope of information was sufficient, but the assimilation of information was ineffective. This was also reflected in the Project story, as there had been instances where decisions had been made but not fully disseminated. The effectiveness of information assimilation increased with project work experience, for example students worked better in the spring semester than in the fall semester.

Since the assessment took place during the spring semester of 2020, the COVID-19 university lockdown had influenced the way the team worked to a great degree. This “tested” the agility of the team in terms of readiness to adjustment and levels of project commitment and value alignment. The team responded to the lockdown by quickly adjusting working habits and increased utilization of online mechanisms for collaboration and were open to adjusting their tasks to make sure that both the project and the theses moved forward.

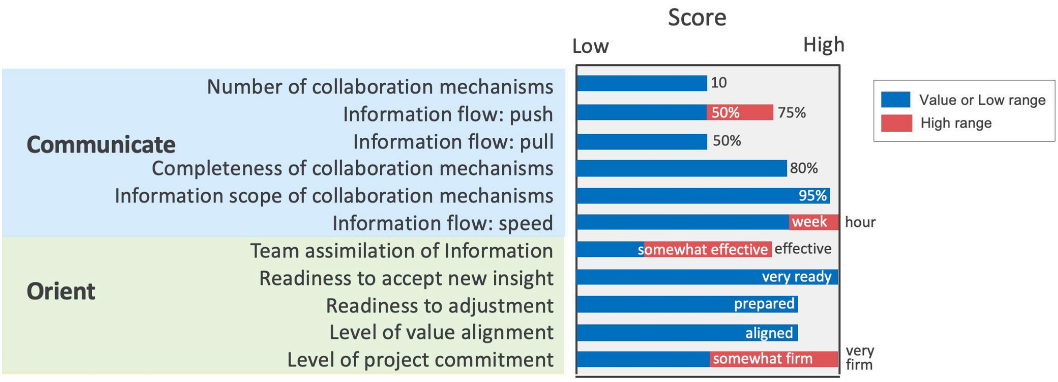


Figure 10. Agile response factors for CubeSat team – communicate and orient.

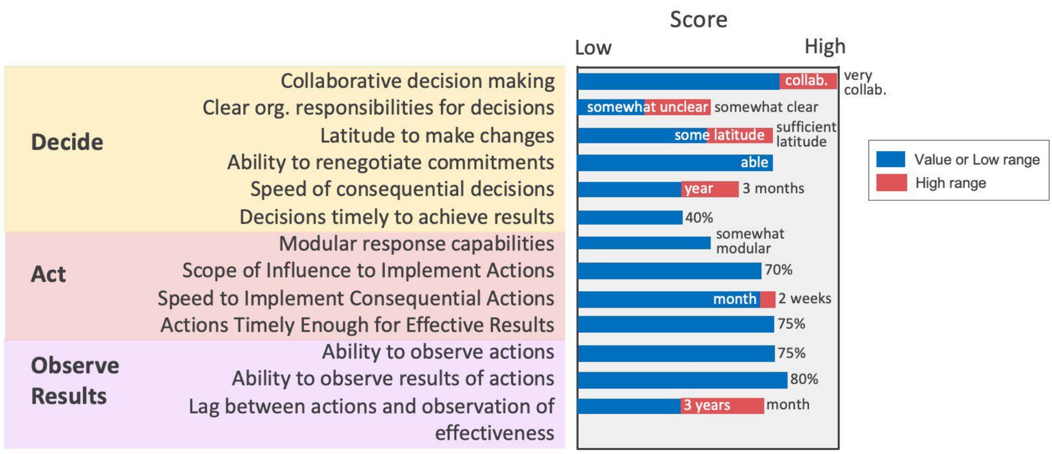


Figure 11. Agile response factors for CubeSat team – decide, act, observe.

Observe, Decide, Act: The team culture is to make decisions collaboratively, especially in the software development, shown in Figure 11. When the decisions influence both software and hardware, it has not been clear who has the responsibility to make a decision and communicate it to the relevant people. There have been instances where decisions were made within a subsystem that did not reach the other affected in a timely manner as explained in the Project story. Although the team has sufficient latitude to make changes, the stakeholders should be consulted before action is taken to ensure that any adjustments do not reduce the mission performance or functionality. Because building the CubeSat and software development takes longer than a single academic year, most of the team members are not able to see the end results of actions taken. Some results have a short feedback cycle, while others will not be apparent until the satellite has been launched and is operational.

Discussion and Conclusion

Based on the above observations, the HYPSO team agreed on factors they would address moving forward:

- Support environment: increase visibility of changes.
- Solution architecture: reduce solution set and architecture dynamics.
- System requirements: increase visibility of changing requirements and reduce dynamics of requirements.
- Information assimilation: increase team assimilation of information.

The *support environment*, shaped by the university context, has historically had the lowest detection rate, largest impact, and highest dynamics. The support environment has had the largest impact on all parts of the system except software development, so anything related to hardware is critical to consider while planning and scheduling the project. There is a need to make the organization more robust towards changes in the support environment, or to anticipate the changes better. However, because it is a university project, it is not straight-forward to improve this factor, for example by vertical integration of functions. We suggest that the team could investigate better goal alignment and organizational structure at the university to improve this factor.

Since the *solution architecture* has been, and is evolving, the adoption of Agile SE system architecture principles around reusability, reconfigurability, and scalability (Dove, et al. 2014) can be used to proactively move to a more agile architecture in future HYPSO projects. The team could benefit from having a continuously updated system model such as the one introduced in the INCOSE Space Systems Working Group (Kaslow, Cahill, Frank 2019), integrated with the other systems such as GitHub and CAD tools. There have been some efforts in this area, but a dedicated effort is needed to create a better knowledge source. If the system model were integrated into the project lifecycle, new team members could use it to enhance their training. The model could also support requirements and with its traceability to components and functions, could also increase the team ownership of the requirements. An improved solution architecture should make the *requirements* management simpler, and the team could also investigate principles for lean requirements management (Oppenheim and Haskins 2016) to minimize the efforts needed.

Two critical aspects were addressed in both the Project story discussion and in the agile response factors assessment, namely, knowledge information and assimilation, and flow of work. While there seemed to be a good coverage of knowledge, and a mix of push and pull information flow, the information did not assimilate well. Moving forward, the team could reach out the INCOSE Social Systems Working Group to explore ways to improve this important communications capability.

A common theme throughout the "Project story" and assessment was the lack of interconnection between the software and hardware teams. While there are documents describing the system, and models showing interdependencies, this way of thinking is not embraced. Traditionally separated in different programs at the university, the students do not have much experience working together across these study areas. It is one of the roles of SE to integrate different disciplines, and the challenge is to do so during the short onboarding period the students have when joining the project for their thesis work. We want to encourage more systems thinking amongst the students using the kick-off and the daily stand-ups events. The hardware, software and operations teams join a joint project stand-up meeting where the systems engineer has the opportunity to encourage interface discussions and dependencies, and to help the students become more aware of their contribution in the bigger system. The project could benefit from having a more systematic approach to integrating hardware and software, and some have suggested having a system model in which all the information is stored, instead of it being separated in design reports for software and hardware.

The team's willingness to adjust to new input indicates a capacity for implementing new agile activities. For example, both software and hardware manage their work using the agile methodology Scrum (Rising and Janoff 2000). This allows for adjustment of plans when new discoveries are made (for example that a laboratory is closed for maintenance, or a critical bug discovered during testing should be fixed quickly), or when people have changing priorities (for example holidays or coursework), using sprints and stand-ups. This agility may be attributed to the team being young, since most of the team is under 30 years.

The ADG method does not specifically measure if an enterprise (a team, project, or organization) is agile or not on some predefined scale of goodness. What it does is help the enterprise understand its ability to thrive in a continuously changing and uncertain environment. The method does this by answering two basic questions; how uncertain and unpredictable the environment of the enterprise is, and how capable is the enterprise in responding to those uncertainties and unpredictabilities. The insight gained from answering these two questions is what helps the enterprise see where improved agility could lead to improved enterprise performance. Once this is known, changes in strategies, tactics, plans and measures can be put into action to improve the agile capability of the enterprise. The ADG method has been structured with an initial set of observation spaces and associated factors to help the user focus on where uncertainties and unpredictabilities are likely to exist. It also provides a way to assess and provide evidence of the relative impact of those uncertainties and unpredictabilities to aide the enterprise in prioritizing improvement actions. The assessment took almost 12 weeks to complete, with 1-1.5 hours meeting each week. This time was used for in-depth discussions and exploration of themes to gain more insight about the organization and was highly valuable.

Using the ADG method was helpful for an academic CubeSat organization because of the discussions and orientations it provided about the team. Some of the questions needed more explanation and refinement to be applicable, and the scoring needed to allow for a range of values to be useful in representing the situation. The latter may also be useful when applied to industrial cases.

References

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Martin R.C., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D., 2001, 'Manifesto for Agile Software Development', <https://agilemanifesto.org/> [accessed 21 October 2020]

Berthoud, L., Swartout, M., Cutler, J., Klumpar, D., Larsen, J.A. and Nielsen, J.D, 2019 'University CubeSat Project Management for Success', 33rd Annual AIAA/USU Conference on Small Satellites, UT (US).

Cho, M. and Mazui, H., 2017 'Best practices for successful lean satellite projects', 4th UNISEC Global Meeting, Varna, Bulgaria.

Dove, R., 1992. 'The 21st Century Manufacturing Enterprise Strategy or What is All This Talk about Agility?' Invited paper originally published by *Paradigm Shift International* (December) and then translated into Japanese and published in a 1993 issue of Prevision, the Japan Management Association Research Institute.

Dove, R. and LaBarge, R., 2014, July. '8.4.1 Fundamentals of Agile Systems Engineering', In *INCOSE International Symposium* (Vol. 24, No. 1, pp. 859-875).

Grande, J., Birkeland, R., et al., 2017. 'Norwegian student satellite program – lessons learned', 68th International Astronautical Congress, Adelaide, Australia.

Kaslow, D., Cahill, P.T. and Frank, R., 2019 'Developing a CubeSat MBSE Reference Model – Interim Status #5', 33rd Annual AIAA/USU Conference on Small Satellites, UT (US).

Lyells, R., Angier, R. and Epps, B., 2018, July. 'Creating Decision Guidance for Applying Agile Systems Engineering', In: *INCOSE International Symposium* (Vol. 28, No.1 pp. 655-670).

Oppenheim, B.W., Haskins, C., 2016, ‘Powerful Opportunities to Improve Program Performance Using Lean Systems Engineering and Lean Program Management’ in: *Advances in Systems Engineering,* Reston, VA (US).
SEBoK contributors, ‘Guide to the Systems Engineering Body of Knowledge (SEBoK)’, SEBoK, 15 May 2020, <https://www.sebokwiki.org/w/index.php?title =Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)&oldid=59187> [accessed 30 September 2020]
Swartout, M., 2019, ‘CubeSat Mission Success: Are We Getting Better?’, 2019 CubeSat Developers Workshop, CA (US).
Radigan, D., 2020, ‘Five agile metrics you won’t hate’ on Atlassian Agile Coach, <https://www.atlassian.com/agile/project-management/metrics> [accessed 11 November 2020]
Rising, L. and Janoff, N.S., 2000, ‘The Scrum software development process for small teams’, *IEEE Software*, Vol. 17, No. 4, pp. 26-32

Appendix: The Project story

The HYPSO project team consists mainly of temporary employees and students at NTNU, with little or no experience with aerospace engineering or multidisciplinary teamwork. The HYPSO project is a part of a larger strategic initiative from the faculty, to be able to deliver fast integration of scientific payloads into small satellites and autonomous assets. The project takes place in a university, and the students mainly do work in the projects through master and bachelor theses. The master students typically arrive in August/September and stay on until May/June the following year. The bachelor students join the team in January and leave in May. The undergraduate students also have course-work in parallel with their theses, and cannot work full-time in the project. The professors supporting the project have not developed systems for space previously, and do not have the time nor knowledge to participate actively in the project. They do, however, provide supervising for the theses and provide direction for the mission. The professors also retain approval responsibility for decisions taken since they are in control of the funding of the project.

The team feels that there aren’t enough people and resources in the project, especially in hardware (both electronics and mechanics). The people working in hardware feel stressed when thinking about the amount of work that has to be done, and how difficult it is to get the full picture. Even though the previous years’ members are available by email and sometimes phone, they feel as though they are not able to understand the full picture of the challenges or the work that remains to be done. The project manager/systems engineer (a PhD fellow) feels this on a personal level when fielding repetitive questions on topics discussed several months ago, such as where to find specific information and how a decision was made. The PM understands that the team members are stressed, and thinks this is one of the reasons that these messages are forgotten/ignored. The PM observes that most of the team members are almost up to speed after 6 months, which is too long when they only have 9 months to contribute to the project in total.

Since the start of the project, there have been challenges in ensuring an understanding of decisions made and sharing these decisions. This first surfaced when some decisions were made, and the person in charge of them was away for a while, and it was not easy for the remaining team members to see why the decision had been made. The team re-iterated the decision with a more standard approach and communicated it in all meetings + documentation. The decision-making process has improved, but there is still a problem of making sure that everyone knows about the decision. The SE has tried setting up a “common model”, but it is not shared or used by all. The SE is also the PM, and a researcher, and feels like there is not enough time to be a good SE. The professors/stakeholders have not been able to do much about this since they have not been able to find someone to fill the role of either PM or SE to replace some of the duties.

Most decisions that must be made are design-decisions and task planning decisions. The design-decisions are implemented in the common model after the decisions have been made, but the common model has not been used to support decision-analysis because there have not been enough resources to develop that capability. Most of the critical (=affecting many subsystems and mission) decisions are made with a minimum number of people, recorded in a minutes-of-meeting, and then documented in the design reports. However, there have been some instances where the team was not able to see the breadth and impact of the decision made, where it has impacted the mission in ways that were not anticipated.

The software part of the project is quite big and involves both pure algorithm development for future enhancement, firmware development for hardware, software interfacing between segments and sub-systems, user interface development, and command & control software of the satellite and payload. It is both a large area and a complex area in the project. An additional complication is that many of the subsystems the payload interfaces with are developed by a different company, with limited documentation and parallel development/changing. There is one person in charge of the software group, who is also working as a researcher.

In March 2020, the team was going to make a go/no-go launch decision based on a Critical Design Review (CDR). The team decided on a set of requirements that would have to be verified by CDR, and if they were not, the decision should be to postpone the launch. Having this deadline helped focus on the development and integration testing, and documentation of these. The process up to CDR uncovered missing requirements and functions, and in some cases, lack of communication between the operations group and software group.

The CDR collocation meeting via Zoom was considered a success by the team. There were good discussions and the team generated valuable feedback across the functional areas. The team was not able to verify all the requirements, and some critical dependability and risk issues were pointed out because of the lack of design maturity. After CDR, some of the momentum was lost. This was partly because of Corona, partly Easter holiday, and partly because students were refocusing on theses writing. This was evident both through lower participation in daily stand-ups, and from their feedback saying that they were focusing on writing their thesis.

Biography

Evelyn Honoré-Livermore. Evelyn is a Ph.D. Fellow at NTNU in the Department of Electronic Systems, researching the integration of systems engineering and project management methods for academic research projects. She is the project manager of the small satellite HYPSO. Evelyn received her MSc in Electronics Engineering from NTNU in 2012, and her MBA from Yonsei University, Seoul in 2017. She has project manager experience working with space systems in the Norwegian aerospace industry.

Ron Lyells. Ron Lyells has been a member of INCOSE for over 12 years. He is current Past President of the Enchantment Chapter and is a co-chair of the Agile Systems & SE Working Group. Retired from Honeywell's Aerospace Group, Ron has over 40 years in various leadership positions involved in product development lifecycle stages ranging from proposal to production support. Other contributions included developing and promoting MBSE techniques and methods, piloting a common system engineering competency framework, individual and team mentoring, and change management facilitation. He holds a B.S degree in Electrical engineering from Arizona State University.

Joseph L. Garett. Joseph received a BSc in Physics and Mathematics from the Ohio State University in 2011 and a Ph.D. in Physics from the University of Maryland in 2017. He now studies hyperspectral imaging and image processing from satellites and drones as a postdoctoral researcher at NTNU.

Robert (Rock) Angier is a retired system engineer and executive IT architect with 44 years of experience at IBM. His areas of contribution include Air Traffic Control collision avoidance, Space Shuttle avionics and flight-to-flight reconfiguration, Space Station, systematic software reuse, and global enterprise business and IT architectures. He has a long-held interest in agile systems. Rock holds an MS in Computer Science with Univ. of Houston, a BS in Physics from Georgia Tech, and is an Associate Fellow with the American Institute of Aeronautics and Astronautics.

Bob Epps has a background in Systems Architecture, Systems Engineering and Software Engineering. Bob worked for Lockheed Martin for 19 years, where he served as Sr. Manager Architecture Integration Technology Center, Sr. Manager Systems Engineering Architecture and Corporate Engineering. Prior to Lockheed Martin, Bob worked at Link Flight Simulation for 25 years in various Engineering and Management positions in Systems Architecture, Systems Engineering, Software Architecture, and IR&D. Bob has a BS degree in Physics from Drexel University and a MS degree in Applied Mathematics from SUNY-Binghamton.

Paper G

Honoré-Livermore, Evelyn; Birkeland, Roger; Haskins, Cecilia, “Addressing the Sustainable Development Goals with a System-of-Systems for Monitoring Arctic Coastal Regions,” INCOSE International Symposium (ISSN 2334-5837), Vol. 30, no. 1, pp. 604–619, 2020 [18]. `http://dx.doi.org/10.1002/j.2334-5837.2020.00743.x`

In this paper we describe how a System-of-Systems approach can support the monitoring of Arctic coastal regions.

30th Annual INCOSE international symposium
Cape Town, South Africa
July 18 - 23, 2020

Addressing the Sustainable Development Goals with a System-of-Systems for Monitoring Arctic Coastal Regions

Evelyn Honoré-Livermore
Norwegian University of Science and Technology
7491 Trondheim
+47 400 18 398
evelyn.livermore@ntnu.no

Roger Birkeland
Norwegian University of Science and Technology
7491 Trondheim
roger.birkeland@ntnu.no

Cecilia Haskins
Norwegian University of Science and Technology
7491 Trondheim
cecilia.haskins@ntnu.no

Abstract. Norway has a large coastal industry and a strong motivation for developing systems to enable sustainable management of ocean resources. Recent advances in collaborating autonomous systems, Internet-of-Things, microsatellites, data fusion, and sensor development have led to initiatives for a more concerted and coordinated effort through the establishment of an ocean studies research project. Applying a System-of-Systems perspective on the project highlights the challenges in terms of interoperability and communication interfaces, as well as revealing the use-cases stakeholders rely on to enable informed decision-making.

Introduction

The United Nations sustainable development goals (UN SDG) are drivers for development activities and national strategies across the world. The Director of the UN Office for Outer Space Affairs (UNOOSA), states that *"close to 40 % of the targets underpinning the 17 UN SDGs rely on the use of space science and technology"*, based on research conducted in 2018 (Pippo 2018). Since water covers 70 % of the planet, it is no surprise that many of the SDG address ocean challenges. *"Understanding the ecology, biogeochemistry and hazards of our oceans in a varying and changing climate is critical to sustaining Earth as a habitable planet"* (IOCCG 2008: p.7).

Developing systems for monitoring the Arctic coastal regions allows decision-makers to develop strategies for sustainable management of these resources. The vastness and challenging environment of these regions mean that it is not cost-effective to base the administration on a single technology for monitoring with the required spatial, spectral, and temporal resolutions. This paper looks at a specific project, from a System-of-Systems (SoS) perspective and describes how it can support the sustainable management of the Arctic coastal regions of Norway.

The MASSIVE (Mission-oriented autonomous systems with small satellites for maritime sensing, surveillance, and communication) is a project funded by the Research Council of Norway (RCN) and the Norwegian University of Science and Technology (NTNU). MASSIVE studies how observations of the ocean can be coordinated between different sensor systems by developing systems to accomplish the goals of effectively monitoring oceanographic phenomena and for distributing data to the scientific community and the relevant decision-makers. It considers small satellites, autonomous vehicles, and both data processing in the sensor nodes and data fusion in operations centers. In the light of MASSIVE's intended capabilities, the question addressed by this paper is: *How can viewing the MASSIVE project as an SoS produce a system that supports the scientific community and informs decision-makers?*

The MASSIVE project concept in Figure 1 gives an overview of included systems and interfaces. The constituent systems (CS) are unmanned aerial vehicles (UAVs), buoys, autonomous surface vehicles (ASVs), autonomous underwater vehicles (AUVs), small satellites (SmallSats), ground station system (GS), and a data processing system. While not shown in the figure, data from monolithic satellite systems such as Copernicus will contribute data to the data processing system. The concept of operation is that the satellite (constellation) will monitor the coast from space, and the autonomous assets from air and on/below the water surface. The operations control center can accumulate and process data collected and provided by the various agents about the ocean.

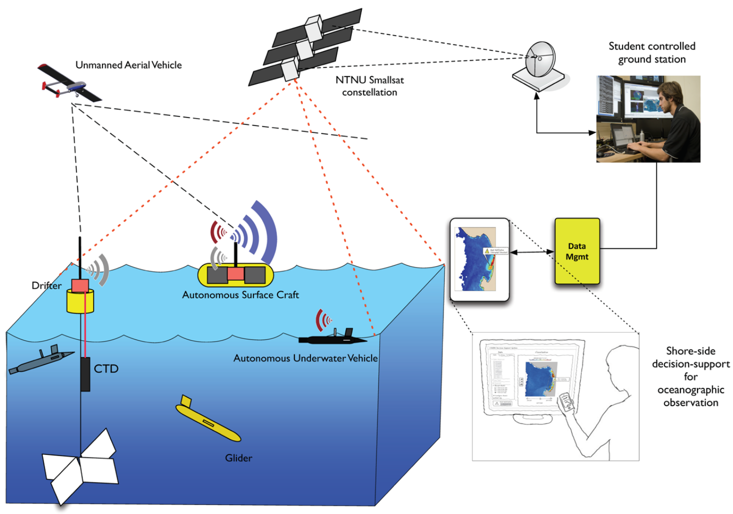


Figure 1. MASSIVE project concept, from (Rajan et al. 2017).

The paper is organized as follows: The first section gives background information on the management of coastal regions and a brief theoretic description of SoS. The next section describes the method used to analyze the MASSIVE project, followed by the analysis and an evaluation of how the MASSIVE SoS can support the scientific community and inform decision-makers in developing strategies for managing Arctic coastal regions.

Background

Managing Coastal Regions

A variety of oceanographic phenomena can be detected with different types of sensors, such as small or large monolithic earth-observing satellites, from ships during scientific cruises, swarms of drones or other autonomous vehicles equipped with sensors, manual tests, physical installations at various points of interest in the region or data gathered as secondary products from other systems. Each of these sensors provides valuable data, but they have characteristics such that no single source can satisfy the needs of the stakeholders.

Norway has a long coastline compared to its population (80,000 km, approx. 5.4 million inhabitants) and a high Gross Domestic Product (GDP) per capita (International Monetary Fund 2019), enabling the government to invest significantly in infrastructure. The Northern coast of Norway has a low population density, making it challenging to rely on human resources to support the surveillance and monitoring needs of the coast. Additionally, the country's industry is mainly offshore oil/gas, fisheries, and aquaculture, which means that the nation has a strong dependence on the coast for sustaining the high standard of living and national income.

The past years have seen an increase in sea temperature and a dramatic loss of ice in the Arctic, leading in part to a rise in ship traffic and a push to explore new oil fields further North. To ensure continued health and viability of the Arctic coastal areas, sustainable monitoring is needed. The need for sustainable monitoring drives the demand for better systems to monitor the Arctic in near real-time so that we can understand the impact of increased human and machine activity on the environment. Furthermore, more activity means a higher risk of loss of life or devices in the Arctic, which is a region underserved by communication and infrastructure, which also poses challenges for search and rescue activities.

RCN has awarded the following research initiatives related to ocean and coast monitoring, signaling how important the coastal areas are for Norway: *"Norwegian Infrastructure for drone-based research, mapping and monitoring in the coastal zone" (7.8M€)*, *"The Norwegian node for the European Multidisciplinary Seafloor and water column Observatory" (7M€)* and *"Ocean Space Field Laboratory Trondheimsfjorden" (18M€)* (Wel 2019).

System-of-Systems

System-of-Systems (SoS) is often used to describe the increasingly complex systems developed today. Maier's definition of an SoS from 1998 is widely cited and is used as a basis for this research. An SoS includes components that are in themselves systems and have operational and managerial independence (Maier 1998*)*. An SoS is distributed, interoperable, and adaptable, and can consist of technical and human components (Madni and Sievers 2016). It is helpful to view the integrated system as an SoS, ensuring the consideration of the whole context when developing the constituents. However, there are additional challenges associated with an SoS which are not present in a system. Firstly, components may reach their own decisions without considering their role in the SoS. Secondly, inherent complexity makes it challenging to model emergent behavior. And thirdly, that testing and verification of the SoS may not be feasible due to its scale and complexity (Madni and Sievers 2016).

Existing systems can be integrated into an SoS, bringing challenges of mismatched interfaces and decentralized operations management (Lindman 2015). Decentralized management creates programmatic problems such as ownership, governance, and data policies. For example, changes to the CS can influence the required capabilities of the SoS and the other CS and requires coordination to manage risk, maintainability, and reliability of the CS and SoS as a whole. An SoS may be a temporary assemblage to satisfy a specific short-term mission or can be adaptable to fulfill a combination of mission objectives that change over time.

There are different types of SoS: virtual (*"…no central management…* (or) *agreed-upon purpose"*); collaborative (*"…interact more or less voluntarily to fulfill agreed upon central purposes")*; acknowledged (*"…independent ownership, objectives, funding, development and sustainment")*; or directed (*"…built and managed to fulfill specific purposes"*) (Madni and Sievers 2016: p.6). The SoS can change or bridge types over time, by adding or removing constituents or if the mission objectives change.

The SoS described in this paper can be classified as something between a collaborative or an acknowledged SoS as the CS have independent management but act together to fulfill the mission objectives.

Analysis Method

The SPADE (Stakeholders, Problem, Alternatives, Decision-making, Evaluation) methodology (Haskins 2008) was applied when analyzing the project. The methodology captures the essential systems engineering principles and can be used continuously at multiple maturity levels of a project. SPADE's focus on stakeholders and analysis of these is relevant when dealing with SDGs, which are so large that there are multiple governmental and private stakeholders involved. This section gives a short description of the method and usage for the case study.

Stakeholders are actors, entities, and anyone affected by the system. They are managed throughout a project's lifecycle, and their involvement can vary continuously depending on the phase (Welford 2018). Stakeholder identification, understanding their level of involvement and contribution, analysis of needs, and management are relevant both to the systems engineer and to the project management. The stakeholders were identified from publications related to the MASSIVE project, and from research news items related to oceanography from RCN. They were assessed according to their interest-influence. The needs from the stakeholders were derived from public documentation review and informal talks with some of the researchers involved in MASSIVE.

The **Problem** definition or description activity is to understand the stakeholders' needs, to uncover the state-of-the-art solutions, and to determine how to measure whether the system solves the problem through metrics of performance and success criteria (Haskins 2008). The problem formulation will vary and change according to the viewpoint taken, the degree of involvement of a stakeholder, and the changes in context from the environment and state-of-the-art development. The context is limited to the MASSIVE project and the Arctic coastal regions, which limits the problem space in which the stakeholders' needs are analyzed. The problem is described from the perspective of oceanographic research and how the MASSIVE project can address the problem by providing new capabilities and information. Specific use-cases were created to contextualize the needs of the stakeholders. This does not rule out future use-cases that expand on the capabilities of the project.

Alternatives are generated based on the different viewpoints from the stakeholder analysis and problem formulation. The alternatives are subject to modification to accommodate the discovery of new options and the changing problem description. The alternatives described are different CS relevant to the overall project, various architectures, and the allocation to meet the system requirements.

Decision-making is a continuous process in a development project, where the people making the decisions determine the quality of the solution chosen (Haskins 2008). It is essential that the decision-making method applied is related to the overall problem formulation and stakeholder analysis, and that it can be tested for validity (Peniwati 2007; Rostaldås et al. 2015). This paper looks at how the project can inform decision-makers for Arctic coastal regions.

Evaluation is key to the whole SPADE framework. Continuous assessment of stakeholders, alternatives, problem formulation, and state-of-the-art solutions allows the project team to adjust the performance metrics and success criteria of the project to meet the changing conditions that arise.

An Analysis of Sustainable Management of the Coast

Stakeholders: Private and Public Stakeholders

The multitude of stakeholders with varying degrees of interest contribute to the SoS complexity. The following stakeholders were identified and categorized according to type and level of influence (Schmeer 1999) as used in previous natural resource studies (Reed et al. 2009; De Lopez 2001). An interest-influence map (Eden and Ackerman 1998) was developed to map the stakeholders and visualize the assessment of the level of influence and interest, shown in Figure 2. While the public has an interest in the sustainable management of the oceans, they are indirect stakeholders represented through ministries (elected officials). The stakeholder analysis to-date has been performed based on a documentation review (Faisandier, Roedler, and Adcock 2019).

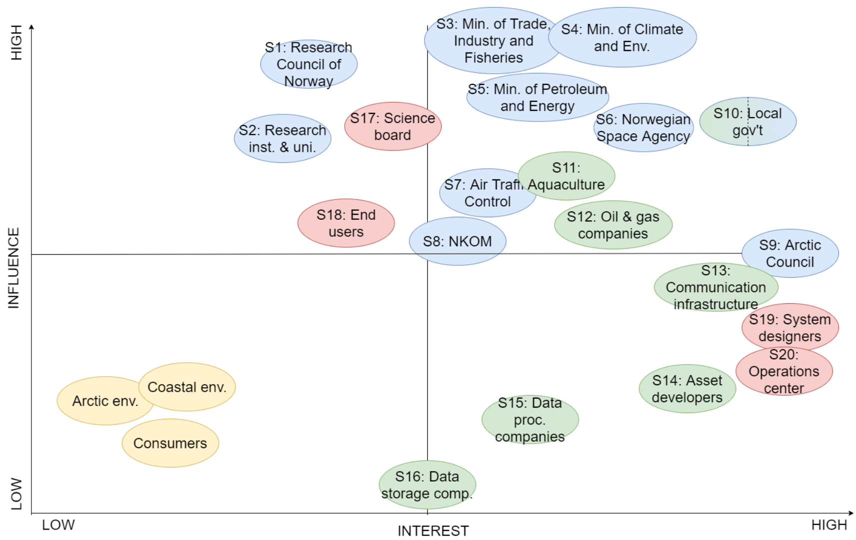


Figure 2. Interest-influence map. NKOM is the Norwegian Communications Authority. Red: MASSIVE; Blue: public; Green: enabling technology; Yellow: passive. Size for readability.

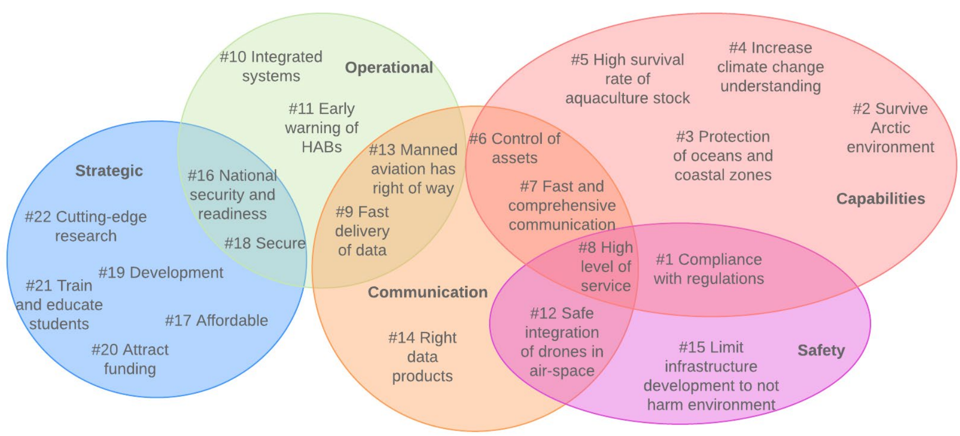


Figure 3. Diagram of stakeholder needs/constraints categorized in Operational, Capabilities, Communication, Safety, and Strategic.

The interest-influence map shows a high concentration of stakeholders in the two right quadrants. The large mass of public bodies in the upper quadrant should move to the bottom quadrant over time as the concept matures, as these have a more substantial influence at the beginning of a project than during the execution. Likewise, the system developers/enablers should move to the upper quadrant in the establishment of the SoS, when these stakeholders have a direct impact on the CS development.

The stakeholder analysis revealed a few high-level needs for an oceanographic monitoring system shown in Figure 3. Cross-mapping of stakeholders and needs are given in Table 1. To measure that the SoS meets the needs, they will be refined and quantified during decomposition into requirements. However, in their current state, they help direct the focus of capability development (upper right corner) while understanding the constraints.

Table 1. A mapping between stakeholder ID (from Figure 2) and needs/constraints (from Figure 3).

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
#1			X	X	X		X	X		X	X	X	X		X	X				
#2									X		X	X		X			X	X	X	
#3	X	X	X	X	X	X			X	X	X	X		X			X		X	
#4	X	X	X	X	X	(X)			X	X	X	X					X		X	
#5			X								X									X
#6							X	X			(X)		X	X					X	X
#7							X	X		X	X	X	X						X	X
#8	X		(X)	(X)	(X)		X	X		X	X	X	X		X	X				X
#9							X	X			X	X	X		X	X				X
#10		X											X	X	X				X	X
#11			X	X	X				X	X	X				X		X	X	X	X
#12	X						X	X		X				X					X	X
#13							X	X					X						X	X
#14			X	X	X						X	X								
#15									X	X			X					X		
#16	X		X	X	X	X	X	X	X	X	X	X	X		X	X			X	X
#17	X									X	X	X		X	X	X		X		
#18	X		X	X	X		X	X	X	X	X	X	X	X	X	X				X
#19	X	X								X				X	X	X			X	X
#20		X	X	X	X									X	X	X			X	X
#21	X	X																	X	X
#22	X	X									X	X		X	X	X			X	X

Problem: Detecting Oceanographic Phenomena

Observing oceanographic phenomena and understanding the ecosystem is complex. While on land, humans can easily see biomass (such as planta and animals), in water microscopic phytoplankton or fish, and sea-mammals hidden under the water are challenging to monitor. This section will discuss the problems associated with detecting oceanographic phenomena and current approaches.

According to the International Ocean Color Coordinating Group (IOCCG), the presence of phyto-plankton is the leading property for understanding the aquatic ecosystems: *"(...) phytoplankton biomass is a key ecological property (...). Ocean-Color Radiometry (OCR) quantifies the base of the* *marine food chain"* (IOCCG 2008: p.7). OCR is used for detection because phytoplankton reflects light. A high spectral resolution provides more biomass information for researchers to better understand the ecosystems and provide information on species type. Algal blooms can move quickly because of ocean currents and having high temporal resolution enables better mapping and understanding of the blooms.

Other phenomena of interest are sea surface temperature (SST), ocean currents, wind data, salinity, sea surface height (SSH), and marine suspended sediments. These phenomena can be viewed in tandem to provide early warning systems for harmful algal blooms (HABs), optimal drilling times for oil and gas operations, safe swimming and diving conditions, stormwater and sewage release able to cause algal blooms, monitoring response of the ecosystem to oil spills, data to optimize competitive sailing paths (e.g. Volvo Ocean Race (IOCCG 1998)) and measure port sea depth.

Sky- and space-based sensors face the challenge of clouds obstructing the view, which can be especially prevalent in the region of interest. A study for the feasibility of optical communication estimated cloud coverage in Norway's Arctic regions, which approximated 25-30% cloud-free days in a year in Arctic land-regions (Bråten and Rytir 2019). The study concluded that there is a lower percentage of cloud-free days over the ocean than over land. On cloudy days, knowledge of oceanographic phenomena and models of how chlorophyll and sediments develop, and move is important to enabling better and timely usage of other sensor systems based on predicted paths.

One of the most significant challenges with the existing systems is that they are not coordinated in what they observe or how. Each system was created with a specific mission or with specific funding but may not have considered other existing or planned systems and how they could cooperate or utilize each other's data to perform the mission. Also, there are significant communication infrastructure challenges with fjords and mountains between the areas of interest, and vast distances to be covered along the Norwegian coastline.

Specific use-cases (UC) were developed as a basis for discussion and to highlight how the MASSIVE concept addresses the needs of the stakeholders. Specific requirements, in addition to the needs in Figure 3, are highlighted.

- **UC-1: Nominal (low resolution) monitoring of the coast (large coverage area).** Requirements: multispectral imaging; medium-scale distributed SST, SSH, salinity, ocean current, and sediment data; edge computing capabilities and low data rate (LDR) OR high data rate (HDR) and ground system computing.
- **UC-2: On-demand high resolution monitoring of HABs (medium coverage area).** Requirements: hyperspectral imaging with high temporal and spatial resolution, plus UC-1.
- **UC-3: Aquaculture monitoring (small coverage area).** Requirements: high frequency oceanographic phenomena monitoring; multispectral imaging; off-board HDR.
- **UC-4: High-resolution monitoring of the coast (various coverage area).** Requirements: high frequency oceanographic phenomena monitoring; hyperspectral imaging with high temporal and spatial resolution. LDR or HDR is dependent on edge computing capabilities.

Constraint: Communication gaps

Most of mainland coastal Norway is covered by mobile communication services such as 4G (LTE, NB-IoT, LTEm) up to some kilometers off the coast. In some deep fjords, there are spots without coverage due to the horizon obstruction and lack of base stations. Satellite services are also available (Iridium, Inmarsat) along the coast, but in narrow fjords, especially GEO-stationary services are limited. Offshore areas south of 70°-75°N can have coverage from GEO satellite services usable for ships, but not usable for smaller platforms/sensors because of the size of the equipment. New satellite solutions such as Norwegian HEO or proposed mega-constellations eventually may offer complementary services (Birkeland and Palma 2018). Around Svalbard, the situation is different from the mainland. Only a small portion of the archipelago has coverage from 4G, limited to areas near Longyearbyen (Telia 2019). A maritime broadband radio network has been tested to provide coverage in central parts (Gulbrandsen et al. 2017). Coverage from geo-stationary systems cannot be relied on for use above 76°N (Plass, Clazzer, and Bekkadal 2015). Thus, much of the Norwegian maritime area, including large parts of the sector above 65°N, is without adequate communication services both for oceanographic research and for Norwegian Search and Rescue (SAR) activities.

UC-1,2,4: Communication with sensors deployed in remote locations: For sensor nodes, several options exist depending on the size of the node, power available, and the amount of data collected (Quintana-Diaz 2019). For sensors with little data (<100 MB/month), systems like Iridium, Argos and OrbComm may provide a solution today. Dial-up Iridium can give a 2.4 kbps link, whereas Iridium SBD, Argos, and OrbComm are message-based systems with message sizes of a few bytes, typically 32 bytes as for Argos. For larger sensors producing more data, there currently is no option to transfer all data over satellite.

UC-3: Communication infrastructure for aquaculture: As aquaculture (fish-farms) move away from the fjords and the coast, the communication systems must move with them. For near-shore installations, custom microwave links can be installed between the shore and the aquaculture site. At the installation site, the network can be distributed through one or several local base stations and provide either specialized data links or other standard communications, such as WiFi and 4/5G. When the distance from the coast increases, satellites may be needed because relaying terrestrial radio signals over long distances and multiple hops offshore is complicated. Inmarsat from GEO-satellites or the upcoming Norwegian HEO-satellites, or services from the proposed mega-constellations, can serve as options if these systems fulfill cost and capacity requirements.

Alternatives: Multi-robot, space-based and ground-based systems

This section will describe different systems that are already in use for coast management, which needs they cover, some of the advantages and disadvantages with the systems, and possible further development needs to satisfy the problem definition. An explanation of the symbols used in the following sections, and of types of ASVs are given in Appendix A.

Multi-robot systems (MRS) consist of different types of robots, such as UAVs, ASVs, and AUVs. An MRS is defined as a system composed of multiple assets where each asset has an individual and a collective task and must have knowledge about the other assets and their movements and performance to achieve the collective mission. There may be multiple MRS' in an SoS, and each MRS can be considered a constituent system.

MRS may be homogeneous (same type of assets with similar characteristics and interfaces) or heterogeneous (combining assets from multiple classes with different interfaces). Much research has been done on both homogeneous and heterogeneous composition and control of assets, as recently discussed in the research and review papers (Birkeland, Zolich, and Palma 2017) and (Zolich et al. 2019). A summary of characteristics is shown in Table 2, where X means that it applies to a range, + means well suited, - not suited to a property assessed.

To utilize MRS to address the use-cases, there are specific communication needs. Drone operators need at least two communication links that could have quite different properties. (1) **The Command & Control (C2) link.** This link will allow the drone to fly beyond-line-of-sight. For this link, controlling the Quality of Service (QoS) is essential. The link must minimize delays, and loss of connection may cause the mission to abort. Iridium provides a basic solution today for the C2-link for some types of flights. Depending on which kind of airspace the drone operates in, Air Traffic Control may require that the operator has a live video feed from the drone to fulfill operations under visual flight rules; hence a broadband link will be needed. (2) **Link for payload data**. This link may not be required for all missions. It will be used for the transmission of payload data, allowing the mission control system to act on payload data during the flight. QoS-requirements for this link may be more relaxed if the data is not critical. In coastal areas near shore, the links can be provided by LTE or 5G, and the mission must be planned according to predicted coverage. Further offshore, satellite systems like proposed mega-constellations could be useful.

Table 2. Unmanned vehicles for coastal and Arctic environments, based on (Zolich et al. 2019). [a]) Depends on wind conditions, it may be difficult to control in strong wind.

	UAV			AUV			ASV		
Type / **Range**	<25 kg	>25 kg	Fixed wings	Light AUV	AUV	Gliders	Renew. energy	Boats	Vessels
0-10 km	X		X	X	X			X	
10-100 km		X	X		X	X	X	X	
>100 km		X	X			X	X		X
Property									
Arctic env.	-	-	+	+	+	+	-	-	-
Precise obs.	++[a]	+	-		+	-	+	-	-
Communication	-	+	+	-	-	-	+	++	++

The ground-based systems are the aquaculture installations, which can host multiple sensors depending on the mass and energy available. These will satisfy many of the UC-3 needs. Other ground-based systems can be buoys with sensors for oceanographic phenomena and a computer with a communication system to interface with other CS. In the Arctic, the challenges are environment and energy for edge computing and data transmission (Quintana-Diaz et al. 2019).

The space segment is dominated by large monolithic communication and by *Earth Observation (EO)* satellites such as the Copernicus program. The Copernicus program supports many of the SDGs, especially when coordinated with a navigation system (UNOOSA 2018). However, the Arctic regions are not addressed as much because of the lack of observation in higher latitudes. There are a growing number of small satellites (<500 kg) and microsatellites (<100 kg) for *EO* and communication. Stratospheric UAVs are a new technology with low maturity that straddles the UAV and space segment. It is expected that payloads on microsatellites today can be deployed eventually on stratospheric UAVs. The cost of a mature stratospheric UAV is not known, but it is expected to be lower than for a monolithic satellite and higher than for a small satellite. A summary of the space segment properties is given in Table 3, where + means suited and - not suited or negative property.

Table 3. Space segment properties. The properties are evaluated in the Arctic context. Payloads are the instruments observing Earth. [a]) Payload properties are not relevant to asses for C2 and datalink. [b]) Spectral availability is related to C2 and payload datalink, not applicable for EO.

	Monolithic satellites			Small satellites			Stratospheric UAVs		
Type / Range	C2	Payload datalink	EO	C2	Payload datalink	EO	C2	Payload datalink	EO
Maturity	+++	+++	+++	++	-	-	---	---	---
Cost	---	---	---	++	++	++	--	--	--
Field-of-view	++	++	++	+	+	+	+	+	+
Payload size > 10 kg	N/A[a])	+++	+++	N/A	-	-	N/A	-	--
Temporal res.	-	-	-	++	++	++	+	+	+
Payload spatial res.	N/A[a])	N/A	++	N/A	N/A	+	N/A	N/A	-
Payload spectral res.	N/A[a])	N/A	+++	N/A	N/A	+	N/A	N/A	+
Spectral avail.	+++	+++	N/A[b])	+	-	N/A	+	-	N/A

Decision-Making

Decision-making for project development is complex because there is managerial and operational independence. Achieving interoperability and ensuring that the right data products are delivered to the end-users so that informed decisions can be reached are the main objectives.

Reasons for viewing the MASSIVE project as a System-of-Systems are twofold:

1. The project team desires to avoid the failure to recognize and benefit from synergies between CS in the solution space such as coordinated ocean observations, and,
2. The number of CS and their communications are too complex to handle as a single system

The stakeholder analysis presented in Figure 3 describes which capabilities the project must provide that the existing CS cannot achieve individually (Axelsson 2015). The current CS have different capabilities and constraints, which must be understood to develop an integrated SoS. Further, the required capabilities given by the stakeholders should be traced to requirements and functions that can be performed by the SoS through decomposition, use-case development, and functional allocation to the different CS, both old and new. Managing an SoS is more complicated than a system because both beneficiary stakeholders and the specific CS stakeholders are involved, sometimes with conflicting expectations.

Looking at MASSIVE as an SoS can increase the understanding of the project management challenges to meet the objectives. This perspective can assist in addressing interoperability and allocation of functions to ensure that the SoS can fulfill the needs. To assess if the MASSIVE project is an SoS, Maier's dimensions were applied to the characteristics of the project in Table 4.

Table 4. The MASSIVE project as a System-of-Systems according to Maier's five dimensions (1998)

Dimension	**Description of MASSIVE**
Operational independence of the elements	Each of the CS are developed to operate independently and can reach decisions without the other elements to perform their own mission objectives.
Managerial independence of the elements	The CS are developed in different phases, and some have higher maturity than others because of this. As an example, the satellite system can be developed and perform independently as a sensor system without the presence of other parts of the MRS.
Evolutionary development	Evolutionary development of the CS allows the SoS' capabilities to evolve with technological advancements, which in turn motivate new capabilities.
Emergent behavior	No single CS can monitor the coastal and Arctic regions with the timeliness and level of detail required without cooperating within the SoS.
Geographical distribution	The developing organizations are not co-located. Also, the CS only interact through information or data exchange and do not rely on physical interactions.

Within each of the CS, there are also decisions to be made, such as energy trade-offs, data budgets, level of autonomy, architecture, and sensor technology. Zolich et al. (2019) discuss possible solutions for the communication infrastructure of heterogeneous multi-robot systems, which are related to the degree of autonomy chosen. However, when the CS are viewed as a part of a larger SoS, the trade-offs become more complicated but may become less complex for technological and architectural decisions. High spectral resolution EO in the space segment could provide more coverage with less cost than equipping all UAVs with high spectral resolution EO. Or, the many small multi-rotor UAVs could carry different sensors for fast response (UC-2, UC-4) while larger fixed-wing UAVs carry several sensors to give an overview (UC-1, UC-3). AUV communication underwater largely relies on the acoustic link to a relay hub which can have a 10-20 km range, but a low bandwidth (<1 kbps). Light UAVs have limited mass available for communication equipment. ASVs can support many communication interfaces depending on the mission (Birkeland, Zolich, and Palma 2017).

The MASSIVE project has focused on developing two of the assets during the first phases; a research ASV called AutoNaut and HYPSO, a small satellite system with a hyperspectral payload. A small satellite designed for Low Earth Orbit (LEO) was chosen because of its low cost, relatively fast development time, and high temporal and spectral resolution. Some stakeholders emphasized that the data collected must be the "right data" and that it is verified. The AutoNaut was chosen because of its multiple onboard scientific instruments for in-situ measurements and can operate autonomously for long periods, which air-borne MRS cannot. While complementary, they are managed and funded separately. Additionally, an operations center is being established that includes command and control of the assets and prediction of oceanographic phenomena by fusing satellite, meteorological, and ocean model data. Future large monolithic satellites may satisfy some of the needs, and new data products may be developed that reduce the need for the MASSIVE entities or be fused to support the more extensive decision-making system.

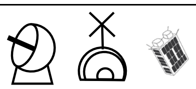
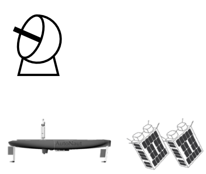
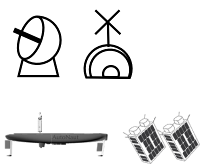
Decision-making for the Arctic coastal region takes place on multiple levels. While there are international committees and directorates concerned with ocean resources, there is no global decision-making body. Decision-making on a multinational scale, a *macro* level, is nearly impossible, and at best tough and time-consuming. At the macro level, agreements between nations can be decided upon, while the actual management of this falls to the lower meso-level. The *meso* level is typically national and local governments and allows for the control of the systems and activities.

Furthermore, at a *micro* level, the local governments can delegate authority to specific companies to perform the actual actions and interactions for making the systems. For example, the local governments (meso) can choose where to build infrastructure for monitoring their harbors, or if a drone-based system should have a deployment site there. The local governments can also act on anyone breaking the local regulations, for example, by having the local police (meso/micro) banning individual shipping companies or fisheries. Providing information may inform decision-makers but does not necessarily result in a structured decision-making process. The political environment and the influential power of the stakeholders affected influence the effectiveness of enacting regulations.

Evaluation

The stakeholder analysis was primarily based on documentation publicly available from the constituent system organizations. Many stakeholders were identified in the first stage of the study, showing the scope of the problem. Not all stakeholders will participate actively in the execution of developing the final solution, but all of them must be allowed to join through gateway reviews or reporting. To strengthen the validity of the analysis, interviews could be conducted with key persons in each of the organizations and other stakeholders. This and other techniques may also uncover needs not expressed in the documentation but relevant to developing the MASSIVE project. For example, the stakeholder list could be expanded based on a recent paper that recommends including peoples whose way of living and observations could contribute to traditional ecological knowledge to help improve marine ecosystem management (Kaiser et al. 2019).

Table 5. Evaluation of how the MASSIVE end-state SoS addresses the specific use-cases.

UC	Systems involved	Evaluation of MASSIVE
All		Requires development of infrastructure and technology sourced both locally and internationally. The project's communication needs will influence infrastructure development. MASSIVE SoS will gather oceanographic data products that can be utilized in understanding climate change. The combination of high temporal, spectral, and spatial resolution through satellite imaging and autonomous asset deployment gives the possibility to gather data cost-effectively. AUVs can be used for fish tracking with optical/radar sensors.
1		**Nominal (low resolution) monitoring of the coast:** The small satellites can be equipped with multi/hyper-spectral imaging sensors. A trade-off must be made between edge computing power requirements and downlink capabilities. Buoys along the coast can provide data on the other oceanographic phenomena.
2		**On-demand high resolution monitoring of HAB:** In addition to UC-1, the ASV can inspect and patrol fjords/coastline where there is some probability that HAB may develop. Will most likely require a small constellation of satellites to provide on-demand monitoring or stratospheric/fixed-wing UAVs equipped for Arctic conditions with multi/hyper-spectral imaging sensors.
3,4		**Aquaculture monitoring, High-resolution monitoring of the coast:** MASSIVE SoS can inform responsible production through better oceanographic data with higher temporal and spectral resolution than existing systems. Requires coordination with end-users to deliver correct data products to inform decisions. Relies on in-situ measurements (ground-based sensors).

The problem description combines aspects from a science community and the Norwegian government. There is not much high-temporal resolution data on oceanographic phenomena available because of the difficulties collecting them. The use-cases selected are based on knowledge of how the CS may interact and to address specific capability needs from Figure 3. An assessment of how the MASSIVE project currently addresses use-cases is shown in Table 5. The use-cases focus the solution work but may also limit the solution space because of specificity. The communication analysis shows that the use-cases are underserved today, and it is difficult to predict when the future systems will be operative. Some initiatives may close these gaps and provide better services.

There should be an aspect of flexibility in the design to address the changing needs of the environment and the development of new technology (Fricke and Schulz 2005), which is more straightforward to assess as an SoS than if it were looked upon as a system. The flexibility can then be built-in through adapting or adding to the CS.

The alternatives listed were limited to autonomous assets because this is the focus of MASSIVE. These CS are being developed in parallel to be integrated or coordinated in the future. It is expected that more MRS will be included, as well as multiple satellites. One of the most challenging aspects is to ensure a good communication infrastructure for the different use-cases and the different CS. Furthermore, the relevance of MASSIVE may change over time.

Conclusion and Future Work

The next decade will be the "United Nations Decade of Ocean Science for Sustainable Development (2021-2030)" (UN General Assembly Resolution 2018). In keeping with the criticality of this topic, this study looked at how an SoS perspective can be used to create solutions that support decision-makers in making informed decisions for the management of Arctic coastal regions. The SoS is challenging to define and describe, to develop, and to test and verify because of the complexity and distributed management of the constituent systems.

Future work on this MASSIVE project should address:

- The sociotechnical aspects of implementing an SoS for monitoring coastal and Arctic regions. On the one hand, there is the technology development and increased infrastructure, which will generate more jobs and a higher level of safety and security. On the other hand, it may be looked upon negatively because it means even more infrastructure in an untouched landscape. Furthermore, the way humans will interface with the SoS, which consists of several assets with varying levels of autonomy, will need to be addressed, for example, avoiding maritime collisions.

- Operational deployment and management of the SoS. Ensuring that the CS are interoperable through agreements and data exchange protocols is critical future work. The SoS viewpoint can be modeled to map out and specify interfaces and to ensure the allocation of functions.

These problems are not specific to the Norwegian Arctic coastal regions and apply to other areas, such as Greenland and Canada, where similar research efforts are underway. Furthermore, the needs will change over time, and technology will develop, supporting the case for the flexibility provided by an SoS perspective. There is an increasing drive for collaboration, cooperation, and interoperability of systems. Different environments give different constraints and performance drivers, and the combination of constituent assets aims to utilize their characteristics to improve the overall solution.

Acknowledgements

This work is supported by the Norwegian Research Council (Grant No. 270959), the Norwegian Space Agency, and the Centre of Autonomous Marine Operations and Systems (NTNU AMOS).

References

Axelsson, J 2015, ‘Systems-of-systems for border-crossing innovation in the digitized society - A strategic research and innovation agenda for Sweden’, *Swedish Institute of Computer Science*, T2015:7, Kista, Sweden.

Birkeland, R & Palma, D 2018, ‘Freely-Drifting Small-Satellite Swarms for Sensor Networks in the Arctic’, London, UK: Springer.

Birkeland, R, Zolich, A & Palma, D 2017, ‘Integrated SmallSats and Unmanned Vehicles for Networking in Remote Locations’, Adelaide, Australia: IAF.

Bråten, LE & Rytir, M 2019, ‘High latitude optical satellite communications - cloud coverage in Norway’, Oslo, Norway: Norwegian Defence Research Establishment (FFI).

De Lopez, TT 2001, ‘Stakeholder Management for Conservation Projects: A Case Study of Ream National Park, Cambodia’, *Environmental Management*, vol. 28, no. 1, pp. 47-60.

Eden, C & Ackerman, F 1998, *Making Strategy: The Journey of Strategic Management*, SAGE Publishing, London.

Faisandier, A, Roedler, G & Adcock, R 2019, ‘Stakeholder Needs and Requirements’. SEBoK Editorial Board. 2019 The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 2.1, RJ Cloutier (Editor in Chief), Hoboken, NJ: The Trustees of the Stevens Institute of Technology, Accessed October 9. <www.sebokwiki.org>. BKCASE is managed and maintained by the Stevens Institute of Technology Systems Engineering Research Center, the International Council on Systems Engineering, and the Institute of Electrical and Electronics Engineers Computer Society.

Fricke, E & Schulz, AP 2005, ‘Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle’, *System Engineering*, vol. 8 no. 4, pp. 342-59.

Gulbrandsen, MU, Stokke, KE, Lie, HW, Høiseth-Gilje, K, Låg, S, Tidemand, N, Skinnemoen, M, Helseth, AM, Ulstein, H & Grünfeld, LA 2017, ‘KVU for elektronisk kommunikasjon i nordområdene’, Menon Economics, Norway, report presented to the government, <https://www.regjeringen.no/contentassets/4bcb1648cd134ff89c454f65864089d8/kvu-for-elektronisk-kommunikasjon-i-nordomradene.pdf>.

Haskins, C 2008, ‘Systems engineering analyzed, synthesized, and applied to sustainable industrial park development’, Ph.D. thesis: NTNU, Trondheim, Norway. 2008:175.

International Monetary Fund 2019, ‘World Economic Outlook Database, April 2019’, Accessed September. <https://www.imf.org/external/index.htm>.

IOCCG 1998, *Minimum Requirements for an Operational Ocean-Colour Sensor for the Open Ocean*, IOCCG, Dartmouth, Canada.

——— 2008. *Why Ocean Colour? The Societal Benefits of Ocean-Colour Technology*, IOCCG: Dartmouth, Canada.

Kaiser, BA, Hoeberechts, M, Maxwell, KH, Eerkes-Medrano, L, Hilmi, N, Safa, A, Horbel, C, Juniper, SK, Roughan, M, Theux Lowen, N, Short, K & Paruru, D 2019, ‘The Importance of Connected Ocean Monitoring Knowledge Systems and Communities’, *Frontiers in Marine Science*, vol. 6, no.309, pp. 1-17.

Fuchs, J & Lindman, N 2015, ‘Using Modeling and Simulation for System of Systems Engineering Applications in the European Space Agency’, in LB Rainey and A Tolk (eds.), *Modeling and Simulation Support for System of Systems Engineering Applications*, John Wiley & Sons, New York, pp. 303-336.

Madni, AM, & Sievers, M 2016, ‘System of Systems Integration: Fundamental Concepts, Challenges and Opportunities’, in R Curran and J Hsu (ed.), *Advances in Systems Engineering*, American Institute of Aeronautics and Astronautics, Reston, VA, pp. 1-34.

Maier, MW 1998, ‘Architecting principles for systems-of-systems’, *Systems Engineering*, vol. 1, no.4, pp. 267-284.

Peniwati, K 2007, ‘Criteria for evaluating group decision-making methods’, *Mathematical and Computer Modelling,* vol. 46, no. 7-8, pp. 935-947.

Pippo, SD 2018, ‘Interview with Simonetta Di Pippo’, *Journal of International Affairs*, School of International and Public Affairs at Columbia University, Accessed September 25. <https://jia.sipa.columbia.edu/space-provides-required-magnitude-perspective-it-unites-us-towards-common-goals>.

Plass, S, Clazzer, F & Bekkadal, F 2015, ‘Current situation and future innovations in arctic communications’, *IEEE 82nd Vehicular Technology Conference,* Boston, MA, pp. 1-7.

Quintana-Diaz, G, Birkeland, R, Ekman, T & Honoré-Livermore, E 2019, ‘An SDR Mission Measuring UHF Signal Propagation and Interference between Small Satellites in LEO and Arctic Sensors’, Logan, UT.

Rajan, K, Johansen, TA, Sørensen, A & Birkeland, R 2017, ‘A roadmap for a SmallSat Program’, Norwegian University of Science and Technology.

Reed, MS, Graves, A, Dandy, N, Posthumus, H, Hubacek, K, Morris, J, Prell, C, Quinn, CH & Stringer, LC 2009, ‘Who's in and why? A typology of stakeholder analysis methods for natural resource management’, *Journal of Environmental Management*, vol. 90, no. 5, pp. 1933-1949.

Rostaldås, A, Pinto, JK, Falster, P & Venkataraman, R 2015, ‘Project Decision Chain’, *Project Management Journal*, vol. 46, no. 4, pp. 6-19.

Schmeer, K 1999, ‘Stakeholder Analysis Guidelines’ in S Scribner and D Brinkerhoff (eds) *Policy Toolkit for Strengthening Health Sector Reform*. Bethesda, MD, Global Health Workforce Alliance.

Telia 2019, ‘Dekningskart (Coverage map)’, Telia, Accessed November 8. <https://www.telia.no/dekning/>.

UN General Assembly Resolution 2018, ‘Oceans and the law of the sea A/RES/72/73’, Accessed November 13. <https://undocs.org/A/RES/72/73>.

UNOOSA 2018, ‘European Global Navigation: Satellite System and Copernicus: Supporting the Sustainable Development Goals’, Vienna: United Nations Office for Outer Space Affairs.

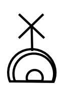
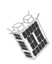
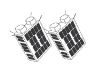
Wel, K 2019, ‘Norsk forskningsinfrastruktur styrkes med én milliard (Norwegian research infrastructure strengthen with 1 billion NOK)’, The Research Council of Norway. Accessed September 29. <https://www.forskningsradet.no/nyheter/2019/norsk-forskningsinfrastruktur-styrkes-med-en-milliard/>.

Welford, J, Scotter, J & Dryden, M 2018, ‘'Picture this!', provoking stakeholder engagement through diagrammatic iteration’, *Systems Engineering Test and Evaluation Conference 2018: Unlocking the Future Through Systems Engineering: SETE 2018,* Melbourne, Australia.

Zolich, A, Palma, D, Kansanen, K, Fjørtoft, K, Sousa, J, Johansson, KH, Jiang, Y, Dong, H & Johansen, TA 2019, ‘Survey on Communication and Networks for Autonomous Marine Systems’, *Journal of Intelligent Robotic Systems*, vol. 95, no. 3, pp. 789-813.

Appendix A

AUV classes: a “Light AUV” is an AUV that can be carried by one man (e.g. <20 kg), “*gliders are long-endurance underwater vehicles”* (Zolich et al 2019). ASV classes: “renew. energy” are often wave and solar-powered, very restricted in power and speed but may have good endurance, “vessels” are large boats/ferries such as a tanker, “boats” are smaller in size, typically up to 10 m.The following icons are used throughout the discussion.

Legend							
							
Ground station	Sensor node	UAV	Fixed-wing UAV	Renew. energy ASV	Monolithic satellite	Small satellite	Small satellite constellation

Biography

Evelyn Honoré-Livermore. Evelyn is a Ph.D. Fellow at NTNU in the Department of Electronic Systems. She is researching systems engineering and project management methods for academic research projects. She is also the project manager of the small satellite HYPSO.

Roger Birkeland. Dr. Roger Birkeland is a post-doctoral researcher at NTNU in the Department of Electronic Systems. He has a Ph.D. in satellite communications from NTNU (2019). He is currently researching small satellite systems and heterogeneous communication systems for remote areas.

Cecilia Haskins. Cecilia is an American living and working in Norway. Technically she has worked in every phase of the software lifecycle and has been a Certified Computer Professional since 1979. Her background includes a BSc in Chemistry from Chestnut Hill College, and an MBA from Wharton, University of Pennsylvania. She has been recognized as a Certified Systems Engineering Professional since 2004 and earned her Ph.D. from NTNU in 2008.

Paper H

Honoré-Livermore, Evelyn; Dallolio, Alberto; Birkeland, Roger; Langer, Dennis D.; Haskins, Cecilia; Johansen, Tor Arne, "MBSE Modeling of a SoS with a Small Satellite and Autonomous Surface Vessels for Persistent Coastal Monitoring," 2021 16th International Conference of System of Systems Engineering (SoSE), 2021, pp. 156-161, `http://dx.doi.org/10.1109/SOSE52739.2021.9497470` [19].

In this paper we go into detail on how a satellite and an autonomous surface vessel can improve monitoring of harmful algal blooms, and how MBSE can support in the design phase of an SoS.

MBSE modeling of a SoS with a small satellite and autonomous surface vessels for persistent coastal monitoring

1st Evelyn Honoré-Livermore
Norwegian University of Science and Technology
Trondheim, Norway
evelyn.livermore@ntnu.no

2nd Alberto Dallolio
Norwegian University of Science and Technology
Trondheim, Norway
alberto.dallolio@ntnu.no

3rd Roger Birkeland
Norwegian University of Science and Technology
Trondheim, Norway
roger.birkeland@ntnu.no

4th Dennis D. Langer
Norwegian University of Science and Technology
Trondheim, Norway
dennis.d.langer@ntnu.no

5th Cecilia Haskins
Norwegian University of Science and Technology
Trondheim, Norway
cecilia.haskins@ntnu.no

6th Tor Arne Johansen
Norwegian University of Science and Technology
Trondheim, Norway
tor.arne.johansen@ntnu.no

***Abstract*—Oceanographic phenomena can be monitored using both remote sensing and in-situ measurements. However, it is challenging to gain actionable insight by just utilizing one source. Combining these data sources in near real-time enables high temporal, spectral and spatial resolution of phenomena in target areas. In this article, we use Model-Based Systems Engineering to model and highlight missing functions or new capabilities needed within an acknowledged System-of-Systems that can support the monitoring of oceanographic phenomena in coastal regions. Different system architectures and a logical architecture have been modeled to provide new insights for developers through reinforcement of a common mental model as well as technical considerations.**

***Index Terms*—systems engineering, system-of-systems, autonomous surface vessels, satellite, remote sensing**

I. Motivation and Background

Monitoring coastal areas and the ocean is necessary to understand the environmental change trends, such as warming of the planet, loss of sea-ice and migrating animal habitats. Human activity is already exploiting and affecting the coastal regions through kelp harvesting, fish farming, offshore oil drilling, shipping, and inadvertently through on-shore operations that influence the ecosystem and atmosphere. There is a need to understand oceanographic phenomena better to allow decision-makers to opt for sustainable choices in the management of the coastal regions. The observation and study of oceanographic phenomena is challenging for several reasons. The regions to be monitored are vast and cannot be monitored with a single class of assets. Moreover, atmospheric and oceanographic phenomena are in continuous fluctuation. Water obscures visibility of sea-mammals, fish and microscopic phytoplankton and no single parameter provides the information many scientists or commercial institutions need.

In this paper we present a System of Systems (SoS) consisting of multiple space and ground assets for monitoring coastal regions for detection of harmful algal blooms. The SoS combines existing assets with new technologies and systems, which results in integration challenges [1], [2]. The management of SoS is more challenging than that of individual systems, especially considering the establishment of unified requirements and capabilities, testing and validation, and the modeling and understanding of emergent behavior of the SoS [3]. We explore how Model-Based Systems Engineering (MBSE) using the Arcadia method [4] can support the design and integration process of an SoS through modeling different system architectures and scenarios, developing logical architectures and discussion points. We address the following research questions:

> *"How can MBSE support the development of an SoS for detection of harmful algal blooms? What insights does the modeling provide?"*

The research reported here is a part of a larger effort at the Norwegian University of Science and Technology (NTNU) to develop and integrate an SoS for consistent monitoring of the oceans using a concert of autonomous agents [5].

A. System-of-Systems for Monitoring Coastal Regions

The system-of-interest consists of multiple Constituent System (CS) such as a ground segment, a space segment and an in-situ segment to satisfy the needs of the stakeholders, requiring an SoS approach to structure the analysis. Managing an SoS is not as straight-forward as managing an individual system, which may already be complex in itself because the

This work is supported by the Norwegian Research Council (Grant No. 270959), the Norwegian Space Center, and the Centre of Autonomous Marine Operations and Systems (NTNU AMOS, grant no. 223254).

SoS involves multiple organizations with different objectives for each of their CS. Using the SoS viewpoint has been applied in other studies for autonomous vehicles [6]–[8]. The SoS viewpoint can aid understanding the emergent behavior the SoS may exhibit depending on the CS and their relationships, especially if the CS choose to "leave" the SoS. Classification of the SoS can be based on the aspect of management, and how it was developed [2], [9]. The types are: (1) Virtual, (2) Collaborative, (3) Acknowledged, and (4) Directed. We classify this SoS as an *acknowledged SoS*, in which the CS "has recognized objectives, a designated manager, and resources (...) [and] changes in the system are based on collaboration between the SoS and the system" [2, p. 6].

The process of decision-making for the design and development of a SoS is more complicated than with a single system. Establishing reliable trade-off models requires insight into the different CS and how their parameters affect the overall achievement of SoS objectives. ISO-21839 outlines different considerations to be made for the life-stages of an SoS; concept, development, production, utilization, retirement, and support [10]. For this paper, we focus on the *concept phase*, exploring viable options and proposing solutions. The specific considerations made are: (1) capability, (2) technical, and (3) management. The proposed CS have constraints, and there are interfaces that should be identified and negotiated early to facilitate adjustment of, or development of, new interfaces and capabilities needed to satisfy the user needs.

B. Monitoring Oceanographic Phenomena in Coastal Regions

In a previous paper we described the high-level design of the SoS and the needs of the stakeholders [11]. Satellite remote sensing has a proven track-record for observing oceanographic phenomena, and most ocean monitoring programs employ either expensive monolithic spacecraft (e.g. the Copernicus program) [12], or data collected via ship-based observations [13]. However, this picture is changing with the advent of small satellites and autonomous vessels. Autonomous systems provide an opportunity for missions in remote or harsh locations, which were previously explored by manned assets [14]. Recent advances in small satellite technology and availability of reliable and efficient Commercial-Off-The-Shelf (COTS) components have enabled faster and cheaper development cycles of science-driven small satellite missions [15].

With the primary focus on monitoring of ocean color, NTNU designed and developed the 6U CubeSat Hyper-Spectral SmallSat for Ocean Observation (HYPSO) [5]. HYPSO is equipped with a Hyper-Spectral (HS) imaging and processing payload that can deliver specialized data products in real time covering a selected geographic region. HS imaging allows for detection and classification of chemical substances based on the reflected spectra. The HS data provided by space assets complemented by geo-physical parameters collected by ground assets, enable marine biologists to study the primary productivity (i.e. plankton and microalgae) of the ocean surface layer as described in [16]. Moreover, the onboard processing payload includes routines for updating software in flight, meaning that new capabilities can be implemented as they are needed, providing flexibility suitable for inclusion in an SoS.

In this paper, we have analyzed the use-case of "On-demand high resolution monitoring of algal blooms" using an Autonomous Surface Vessel (ASV), the NTNU AutoNaut [17], [18], and HYPSO. The commercially available, wave- and solar-powered AutoNaut is equipped with a scientific sensor suite and can operate autonomously in both coastal regions and open ocean. The sensors sample upper water column properties, such as ocean currents, water conductivity, temperature, salinity, oxygen saturation, chlorophyll, organic matter, as well as atmospheric parameters.

For the purpose of this work, we define high resolution on-demand monitoring as:

- High temporal resolution: revisit times less than 3 hours because the algal blooms are dynamic and can move and change characteristics quickly [5], [19].
- High spectral resolution: more than 20 spectral bands in the visual spectrum are required to identify different phytoplankton and other colorizing phenomena [20], [21].
- Upper water column sampling of multiple characteristics such as sea surface temperature, salinity, oxygen concentration, chlorophyll concentration, wave height and weather conditions [20].

II. Modeling a System-of-Systems

Modeling complicated systems to gain pertinent knowledge for design and decision-making can be done with different methods and tools in the various life-stages of the SoS [22]. Firstly, modeling capabilities and objectives of the SoS, as well as different concepts of operations are required to determine the SoS architecture options via a top-down modeling approach. Secondly, modeling is needed for each of the CS to determine and define the interfaces and how the CS satisfy the objectives of the SoS. Thirdly, the modeling should support simulation of or prediction of emergent behavior, to lower the probability of undesired effects. Lastly, modeling for testing and validation of the SoS should ensure the traceability from the top-level objectives to the lower-level requirements and functional elements of the CS.

The process of exploring the solutions in this paper have been supported by the use of Capella 1.4.1[1] with the Arcadia method. This has facilitated discussions in the project development team in addition to providing specific functional scenarios and functional chains. The Arcadia method looks at *Operational analysis* in which stakeholder needs, the environment, actors and activities are defined; *System analysis* in which the boundary and context of the system are defined, and behavior modeling of what the system must accomplish; *Logical architecture* in which the system is seen as a white box and functions are allocated to different logical components in order to fulfill the expectations; *Physical architecture* in which the physical architecture describes how the system will

[1] Open source system MBSE tool https://www.eclipse.org/capella/

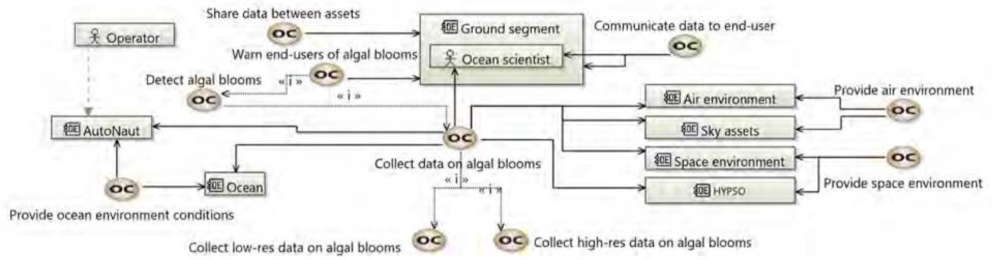


Fig. 1. Operational capabilities. OC = Operational Capabilities. $<< i >>$ means an included capability. Dashed line = communication link. Solid line = involved operational elements.

be built; and, *Product breakdown structure* defining physical components or configuration items that are in the system in its realization.

The use of MBSE has gained strong adoption the last decades, supported by the establishment of SysML and the development of software tools that support MBSE. Using MBSE reduces some of the challenges with document-based systems engineering, by allowing different viewpoints to show relevant information of the same system without needing to continuously update and trace documents [23], [24]. For this study, we have used the diagrams and artifacts available in the three high-level viewpoints in Capella. The Arcadia method does not specify which level to start with, and Capella allows for semantic referencing between elements at each level. This enables iteration and designing with agility at both system and logical level as we learn more about the systems, user needs, and constraints.

A challenge that emerges when using conventional MBSE for modeling SoS is related to the choice of the "system-of-interest", since there are multiple CS which are all system-of-interests at the same time but to different stakeholders. Furthermore, the architecture can quickly become complicated, and modeling should allow for "sufficient requisite variety, parsimony and harmony [25, Table 2]."

III. Results and discussion

The purpose of the modeling efforts was to map out the capabilities required to meet the use-case needs, describe the technical considerations such as interface design, sensor limitations or communication constraints, and identify management considerations. All diagrams shown are from Capella, and are representations of the system model that has been developed using the Arcadia method.

A. Operational analysis

The operational analysis identified the actors and entities involved, i.e. operators, scientists, space environment and ocean environment, with associated *operational capabilities (OC)* as shown in Fig. 1. The central *operational capability*, *"OC: Collect data on algal blooms"* includes other capabilities such as *"OC: Detect algal blooms"*, and is also split into collecting both high and low resolution (spectral and temporal) data on algal blooms. Low resolution data could increase the coverage area or reduce the size and speed of the data link required by the asset collecting data. This separation is to show that the system design may differ for each of the capabilities, and that the detection of algal blooms is a capability that will be offered in the future because it is dependent on more functions and parameters.

Next, *operational activities (OA)* were identified and placed in an operational context with the entities and actors. For example, the ocean will act as both an environment and a data source, and the AutoNaut needs to be *"OA: Protected against ocean environment"* to survive in addition to collecting samples. Not all activities, actors or entities identified in the operational analysis phase need to be transitioned to "lower level" analysis, as some may be provided by a COTS provider, or identified later in the development life cycle. While the specific requirements had not been derived at this stage, it was possible to model the *OA* of the *Actor Ocean Scientist* by keeping the description at a higher abstraction level, e.g. *"OA: Ask for data in specific area"*. The system model can be continuously refined, and having the requirements before the modeling starts is not necessary. Similarly for the data format or details of exchanged information.

B. System analysis

The system analysis in Capella resulted in three *exchange scenarios* which would guide the rest of the domain-specific analyses (e.g. coverage area and communication analyses) and would then feed back to the system design. The three scenarios were: Scenario 1: using existing satellite databases to provide the AutoNaut with instructions on where to perform in-situ measurements; Scenario 2: using processed data received through the HYPSO ground segment to inform where the AutoNaut should measure; and (3), a special case where HYPSO could communicate directly with AutoNaut using a dedicated communication interface, nicknamed AutoSat. There is a "master exchange scenario" diagram to show which ones can be chosen to provide the end user with required information, Fig. 2.

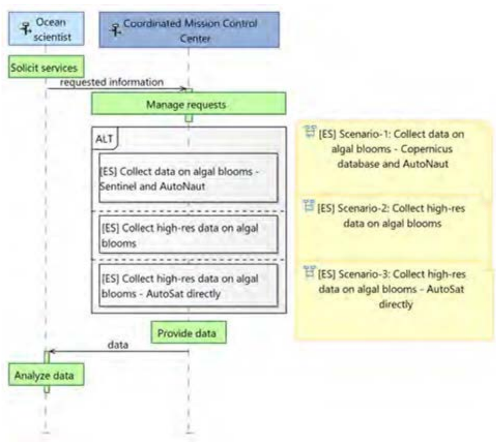


Fig. 2. Modeling choice between scenarios. The yellow sticky-notes are links in the Capella software.

The different system functions involved are also represented by *functional chains* in the *system architecture blank* diagram in Fig. 3. The functional chains can later be broken down and can aid verification and validation activities of the SoS, by highlighting what the developers should be testing to ensure that the scenarios can be fulfilled.

C. Logical analysis

The logical analysis was mainly used to map the functions to different *logical components*, such as the AutoNaut processing system or the ground processing system. From the system needs analysis, a *logical architecture blank* diagram was developed with the required logical functions needed to fulfil the system functions. There is model consistency through automated transitions of actors and functions, and allocations of these are shown in Fig. 4.

The *Ocean scientist actor* functions include "Define algal parameters" and "Set location", which are the critical functions needed to manage the assets. However, we can expect that the *Ocean scientist actor* will have more functions, but these are not relevant for the current discussion. The choice of which elements to display in a diagram at any time without losing information in the system model can greatly help discussion by managing the requisite variety, parsimony and harmony.

D. Insight provided by modeling

The **capabilities** needed were mapped out in the *operational analysis*, which can be further elaborated with e.g. "Operational Activity Interaction" diagrams. Use-case diagrams, such as in SysML, could also have been used to identify needed capabilities. The capabilities and operational activities may be further allocated to functions that can be verified, and associated requirements. The system model maintains the semantic relationships between operational needs, activities, system functions, logical functions, etc., which could be more complicated to express and maintain consistency of across documents. The system model gives the system context and scenarios, which, when supplemented by textual requirements gives a holistic and rich picture of the state of the system [4].

Technical considerations were discussed in both the system level analysis and in the logical level analysis. We found that the exchange of information could happen in three different scenarios. Elaboration of exchange scenarios also identified missing system functions and the need for better coordination of CS development efforts. This coordination entailed agreement on the data to be exchanged, documentation of the technical specification for the communication system, and analysis of the impact of the interface on the collective data budget for the SoS. The discussions leading up to the (relatively) simple logical architecture identified the need to develop a function that could choose the communication system which determines which of the scenarios would be selected. Furthermore, a "Coordinated Mission Control Center" was identified as a required logical component, to coordinate the different CS involved and their capabilities to fulfill the needs of the end-users.

Critical **management considerations** were not uncovered during the modeling process. This may be because the CS are under the same operational management (in the case of HYPSO and the AutoNaut), or because they are provided as a service (such as ground segment and Copernicus data), or that this system model and MBSE approach do not incorporate these aspects well enough to give insight. However, there is an important managerial consideration to be made when it comes to *willingness-to-pay* for a potential **"Harmful Algal Bloom Watch"** service not yet shown or allocated. The AutoNaut makes use of commercial communication services such as Iridium, and the satellite needs a ground segment that can support both large and small data volumes, which may be costly. For research institutes, the specific requirements and end-users may not be actively involved in the SoS development, but represented by reviewing research in the specific field of interest. In this context, the MBSE approach with high-level needs represented by operational activity and capability elements allows the researchers developing the CS and SoS to be aware of the existence of needs, and to account for them until they evolve to specific requirements.

E. Lessons learned and future modeling

We chose to use the Capella tool because it is open source, supports integration with GitHub, has a very active user group on forums, and multiple webinars that can be used for training, lowering the barriers for usage. While the online resources can help the users get familiar with the tool, time and resources are still required to use it effectively. We found that using webinars and examples that closely resemble the system-of-interest were helpful to understand how to start the modeling effort.

Capella provides progress flags such as "to be reviewed" or "draft" that can be attached to all elements to assist the development process. The progress monitoring can be viewed

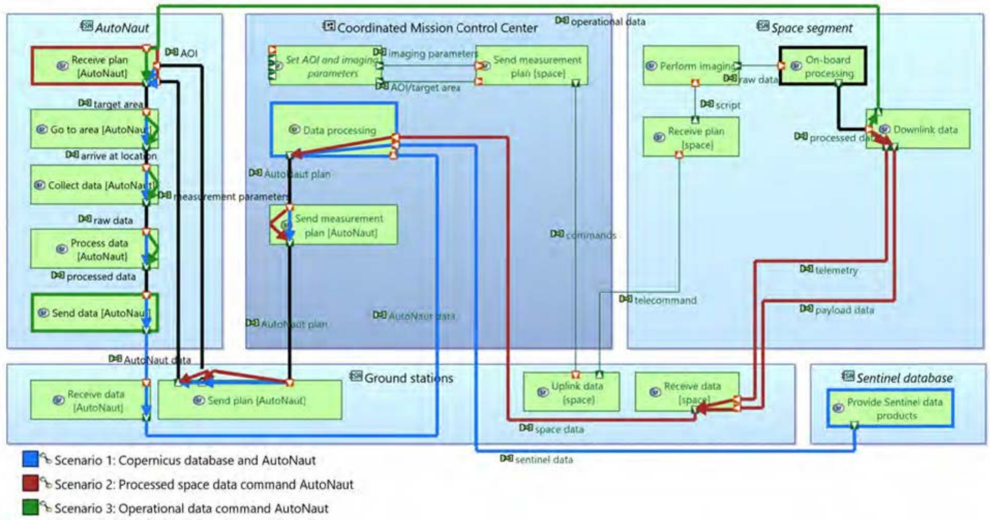


Fig. 3. System architecture blank with functional chains. The blue functional chain includes the system functions for Scenario 1, the red functional chain for Scenario 2, while the green for Scenario 3. The black is when more than one functional chain involves those exchanges.

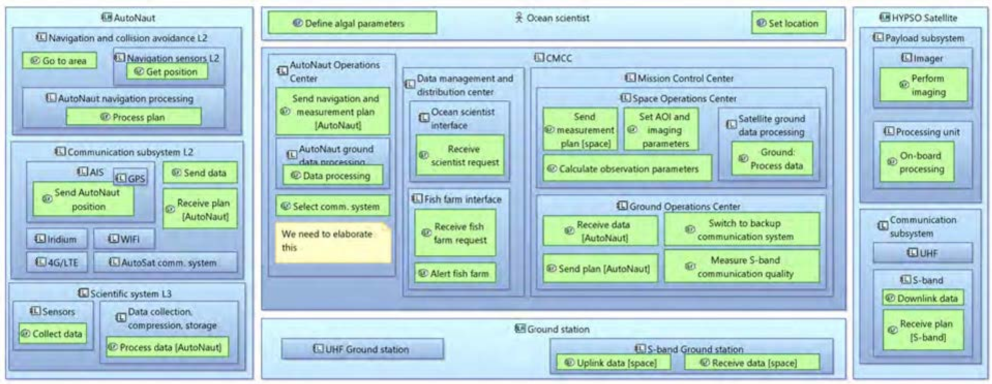


Fig. 4. Logical architecture blank diagram. The exchanges are not shown because it would make the diagram messy, but are available in the system model. Sticky notes are included for highlight where more development is needed.

and exported so that the system engineers of the different CS and the SoS coordinator have visibility of the status of the development. Capella also has built-in model validation in terms of: integrity, design completeness, design coverage, and traceability. Designers can also specify their own rules than can be executed on the model.

It is challenging to train the systems engineers in SoS and MBSE [26], and to engage the CS developers in providing necessary details to build a useful system model. One reason for this is that the purpose of the modeling effort and expected insights are not clearly defined at the onset of the effort, and the CS developers do not know what information is needed or to what level of detail. For the HYPSO and AutoNaut developers, the operational diagrams and exchange scenarios helped them understand what information was needed to give valuable insight. Moreover, what is needed to document the system model sufficiently so that it can be re-used. While SoS as a concept is not new, thinking in terms of SoS engineering instead of "just" Systems Engineering (SE) [27] supported the SoS development because the designers were using appropriate terms. For example, *operational capabilities* instead of specific *system requirements*. It is also more complicated to deliver a resilient SoS with consistent performance to the stakeholders. Future modeling should look at multi-level risk analysis and resilience, to avoid adverse effects to the SoS if one CS leaves the SoS, or is compromised by e.g. cybersecurity issues.

Furthermore, the management considerations should be explored further. This includes creating high-level plans for integration and updating of the SoS, aligning funding for implementation of necessary interfaces, synchronizing testing, and continuous risk management for development and operations. Using systemigrams [22] have been recommended for conceptualizing complexity in SoS, and can be used to complement the analysis in Capella, and can provide new insights of the interdependencies and sociotechnical aspects.

IV. Conclusion

To understand our oceans we need to use a variety of sensing instruments and assets. Oceanographic phenomena present spatial and temporal scales that can vary significantly, e.g. algal blooms span over meso-scale ranges whereas primary productivity happens at microscopic scales. The employment of space-borne sensors in combination with in-situ measurements provided by ASV, allows ocean scientists to gain new insight about coastal regions and about the effects of environmental changes. However, most information is obtained by coordinated measurements and data processing, requiring an SoS approach.

In this paper we have described how Model-Based Systems Engineering can assist system developers in aligning their efforts by specifying models with the capabilities needed by stakeholders. Operational analysis, system analysis and logical analysis have provided both capability identification, important technical considerations for further integration and development of CS, but not management considerations. The modeling effort was limited to what was needed for the CS development in at the current phase. Future modeling efforts are focused on developing integration and other processes to use results from the domain-specific tools to support system trade-offs and validation and verification activities.

Acknowledgment

The authors thank the NTNU SmallSat Lab and NTNU UAV-lab for the work and cooperation in this development. E.H-L. thanks Stéphane Lecrampe (ObeoSoft Canada Inc) for modeling ideas and support.

References

[1] J. Fuchs and N. Lindman, "Using modeling and simulation for system of systems engineering applications in the european space agency," in *Modeling and Simulation Support for System of Systems Engineering Applications*, L. B. Rainey and A. Tolk, Eds. New York: John Wiley and Sons, 2015, pp. 303–336.

[2] A. M. Madni and M. Sievers, "System of systems integration: Fundamental concepts, challenges and opportunities," in *Advances in Systems Engineering*, J. Hsu and R. Curran, Eds. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2016, vol. 252, pp. 1–34.

[3] INCOSE, "Incose Systems of Systems Primer INCOSE-TP-2018-003-01.0," Tech. Rep., 2018.

[4] S. Bonnet, J.-L. Voirin, and J. Navas, "Augmenting requirements with models to improve the articulation between system engineering levels and optimize v&v practices," *INCOSE International Symposium*, vol. 29, no. 1, pp. 1018–1033, 2019.

[5] M. E. Grøtte, R. Birkeland, E. Honoré-Livermore, S. Bakken, J. L. Garrett, E. F. Prentice, F. Sigernes, J. T. Gravdal, and T. A. Johansen, "Ocean Color Hyperspectral Remote Sensing with High Resolution and Low Latency – the HYPSO-1 CubeSat Mission," *IEEE Transactions on Geoscience and Remote Sensing*, 2021, to be published.

[6] R. Gomes, J. Straub, A. Jones, J. Morgan, S. Tipparach, A. Sletten, K. W. Kim, D. Loegering, N. Feikema, K. Dayananda, G. Miryala, A. Gass, K. Setterstrom, J. Mischel, D. Shipman, and C. Nazzaro, "An interconnected network of UAS as a system-of-systems," in *IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, 2017, pp. 1–7.

[7] R. Zhang, B. Song, Y. Pei, and Q. Yun, "Improved method for subsystems performance trade-off in system-of-systems oriented design of uav swarms," *Journal of Systems Engineering and Electronics*, vol. 30, pp. 720–737, 2019.

[8] M. LaSorda, J. M. Borky, and R. M. Sega, "Model-based architecture and programmatic optimization for satellite system-of-systems architectures," *System Engineering*, vol. 21, 2018.

[9] M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[10] International Standards Organization 2019, "ISO21839:2019(E) systems and software engineering - System of systems (SoS) considerations in life cycle stages of a system," 2019.

[11] E. Honoré-Livermore, R. Birkeland, and C. Haskins, "Addressing the sustainable development goals with a system-of-systems for monitoring arctic coastal regions," in *INCOSE International Symposium*, vol. 30, 2020, pp. 604–619.

[12] M. Aguirre, B. Berruti, J.-L. Bezy, M. Drinkwater, F. Heliere, U. Klein, C. Mavrocordatos, P. Silvestrin, B. Greco, and J. Benveniste, "Sentinel-3 - the ocean and medium-resolution land mission for GMES operational services," *ESA Bulletin*, vol. 131, pp. 24–29, Aug 2007.

[13] S. R. Smith *et al.*, "Ship-based contributions to global ocean, weather, and climate observing systems," *Frontiers in Marine Science*, vol. 6, p. 434, 2019.

[14] A. Zolich, D. Palma, K. Kansanen, K. Fjørtoft, J. Sousa, K. H. Johansson, Y. Jiang, H. Dong, and T. A. Johansen, "Survey on communication and networks for autonomous marine systems," *Journal of Intelligent Robotic Systems*, vol. 95, no. 3, pp. 789–813, 2019.

[15] M. N. Sweeting, "Modern small satellites-changing the economics of space," *Proceedings of the IEEE*, vol. 106, no. 3, pp. 343–361, 2018.

[16] IOCCG (2014), *Phytoplankton Functional Types from Space*, ser. Reports of the International Ocean-Colour Coordinating Group, No. 15, IOCCG, S. Sathyendranath, Ed. Dartmouth, Canada: IOCCG, 2014.

[17] A. Dallolio, B. Agdal, A. Zolich, J. A. Alfredsen, and T. A. Johansen, "Long-endurance green energy autonomous surface vehicle control architecture," in *OCEANS 2019, Seattle, Washington.*, 2019.

[18] NTNU, "Ntnu AutoNaut documentation wiki," http://autonaut.itk.ntnu.no, Dec 2018.

[19] T. D. Dickey and R. R. Bidigare, "Interdisciplinary oceanographic observations: the wave of the future," *Scientia Marina*, vol. 69, pp. 23–42, 2005.

[20] IOCCG, *Why Ocean Colour? The Societal Benefits of Ocean-Colour Technology*, ser. Reports of the International Ocean Colour Coordinating Group. Dartmouth, Canada: IOCCG, 2008, vol. 7.

[21] F. Ortenberg, *Hyperspectral Sensor Characteristics*. Hyperspectral Remote Sensing of Vegetation, 10 2011, pp. 39–68.

[22] B. Sauser and J. Boardman, "Systemigram modeling for contextualizing complexity in system of systems," in *Modeling and Simulation Support for System of Systems Engineering Applications*, L. B. R. Tolk and Andreas, Eds. John Wiley and Sons, 2015.

[23] T. Huldt and I. Stenius, "State-of-practice survey of model-based systems engineering," *Systems Engineering*, vol. 22, no. 2, pp. 134–145, 2019.

[24] K. Henderson and A. Salado, "Value and benefits of model-based systems engineering (MBSE): Evidence from the literature," *Systems Engineering*, pp. 1–16, 2020.

[25] J. Ring and A. M. Madni, "Key challenges and opportunities in 'system of systems' engineering," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, 2005, pp. 973–978.

[26] J. Holt, S. Perry, R. Payne, J. Bryans, S. Hallerstede, and F. O. Hansen, "A model-based approach for requirements engineering for systems of systems," *IEEE Systems Journal*, vol. 9, no. 1, pp. 252–262, 2015.

[27] J. Dahmann, G. Rebovich, J. Lane, R. Lowry, and K. Baldwin, "An implementers' view of systems engineering for systems of systems," in *2011 IEEE International Systems Conference*, 2011, pp. 212–217.

Paper I

Honoré-Livermore, Evelyn; Fossum, Knut Robert; Veitch, Erik, "Academics' Perception of Systems Engineering and Applied Research Projects," Systems Engineering, (The Journal of The International Council on Systems Engineering), Wiley Periodicals, 2021, pp. 1–16, `http://dx.doi.org/10.1002/sys.21599` [20].

In this paper we report from 18 semi-structured interviews and show academics' perception of SE and PM.

Received: 30 November 2020 | Revised: 28 June 2021 | Accepted: 2 August 2021

DOI: 10.1002/sys.21599

REGULAR PAPER

WILEY

Academics' perception of systems engineering and applied research projects

Evelyn Honoré-Livermore[1] | Knut Robert Fossum[2] | Erik Veitch[3]

[1] Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway

[2] Centre for Interdisciplinary Research in Space (CIRiS), NTNU Samfunnsforskning, Trondheim, Norway

[3] Department of Design, Norwegian University of Science and Technology, Trondheim, Norway

Correspondence
Evelyn Honoré-Livermore, Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim 7491, Norway.
Email: evelyn.livermore@ntnu.no

Funding information
Norges Forskningsråd, Grant/Award Numbers: 223254, 270959

Abstract

There is an increased complexity in applied research projects that demand more researcher skills, especially in managing the research project and interdisciplinary work. Researchers receive little training in how to manage such projects, yet most manage to deliver project results. There is a tradition of project management and systems engineering which benefits complex development projects in industrial settings. Despite the apparent benefits, we found limited application of either project management or systems engineering practices in academia. Furthermore, we found barriers to applying these practices in the first place, such as a lack of clear guidance or tools for their execution. A case study based on 18 semi-structured interviews provides a perspective on academic research projects, and how the application of project management and systems engineering in an academic setting shows promise to improve the realization of concept design.

KEYWORDS
project management, projects, research, systems engineering

1 | MULTIDISCIPLINARY COMPLEX RESEARCH PROJECTS IN ACADEMIA

Universities and research institutes are increasingly asked to participate in applied research and development activities, sometimes in parallel with basic research; adding *engineering tasks* to their responsibilities in addition to research and education.[1] The motivation for conducting this analysis was to understand how a university executes projects concurrently with research, how academic staff view projects, and what opportunities exist for improving the system to support researchers in balancing the workload of performing in these different roles.

Research activities, especially in technology-related fields, sometimes need advanced infrastructure and multidisciplinary cyber-physical systems. For example, in the field of cybernetics, we observed that much of the prior research was focused on algorithm development and simulation to increase autonomy at some research institutions. Today, there is an increased focus on algorithm development, simulation *and* implementation and testing of these in a sociotechnical setting, for instance on drones or other types of multi-robot systems interacting in society.[2] This complexity creates continuity challenges to pick up the research where a colleague left off and continue pushing the research frontier. In academia, the non-tenured staff is constantly in flux, depending on the research funding structure, resulting in a dynamic research environment. Concurrently, the research activities are funded through *projects*,[1] requiring scientific personnel to act as *project managers* to manage the funding applications, financial reporting, and research tasks. However, not all universities researchers provide training in managing projects or engineering tasks, leading to cost and schedule overruns, or under-delivery. It may also lead to frustration, stress, or conflict within the research teams: challenges for which the *project managers researchers* may not have the training to recognize or remedy.

Industry, such as aerospace, defence, or pharmaceutical companies, is structured to do project work with the needed personnel, training, processes, standards, supplier chains, workshops, and other support systems in place. Naturally, the industry faces its own set of challenges, suggesting that there may be opportunities for learning between the two areas. Like Google, Tesla, or SpaceX, some companies do research and development and deliver cutting-edge products and services simultaneously, often while working together with universities to push the research frontier.[3] Distinctions between university and industrial research projects may be explained by known challenges of combining efficient resource management with high-level exploratory knowledge work[4–6] and differences in established norms, ideals and identities characteristic of the academic researchers and their scientific practice. Such distinctions support the motivation for studying the structure and systems associated with Systems Engineering (SE) and Project Management (PM) in academia and research institutions.

The importance of the coordination and integration of SE and PM tasks and roles has gained attention recently. The recognition of overlapping artifacts, tasks, processes, and responsibilities such as *risk management*, *planning*, *configuration management*, *data management*, *assessment*, *customer interaction*, and *decision analysis*[7–9] has been shown to increase the probability of success of project results for cost, technical aspects, and schedule performance.[10–13] Similarly, if tailored to the situation, for example, collaborative research projects, existing PM knowledge can reduce the time required to learn-by-doing and draw from the various benefits of a professional and targeted project manager.[14] While PM is traditionally concerned with cost, schedule, and scope aspects, and SE with the product aspect, these are not necessarily easy to separate. Studies suggest that coordinating SE and PM will benefit most types of projects and organizations.[15]

Observations gathered during action research of involving two case studies, where researchers had to balance engineering, management, education, and research tasks, without much formal training in SE or PM, suggested a research focus and the questions we sought to address were as follows:

- **RQ-1**: How can an engineering project ensure the fulfillment of academic research goals (in a university setting)?
- **RQ-2**: How can engineering goals and individual research goals be fulfilled simultaneously?
- **RQ-3**: How do researchers understand SE and PM?

To address these research questions, we examined two groups working on externally funded projects to develop space technology. The data collected is based on 18 semi-structured interviews of between 45 and 60 min.

This paper is structured as follows: In Section 2, we describe SE and the differing perspectives of hard and soft systems. Following that, we describe what a research project and process are, and an introduction to the integration of SE and PM. In Section 3, we outline the research methodology and analysis method applied. Next, we report on the results in Section 4, and discuss what this means for research projects in Section 5. Finally, this paper concludes with a set of additional questions for the organization of future research projects.

2 | BACKGROUND

A research organization is a sociotechnical system developing the *engineered* systems. This sociotechnical system needs to be analyzed and understood so that we can improve the way we develop systems. In this section, we outline SE and sociotechnical research. We then describe the integration of SE and PM, since these fields offer heuristics for managing complex projects. Finally, we introduce applied research projects and their role in academia.

2.1 | Systems engineering

SE is concerned with understanding the needs of the stakeholders and the context of the problem and determining how to meet those needs with a system or product throughout its useful life.[16] SE emerged as a discipline during the Apollo program, where it became clear that the current working practices were not adequate to manage the complexity of putting a man on the moon and returning him back safely.[17] The discipline and practices have evolved and been refined, but the essence remains the same.

SE can be viewed as a methodology; a set of methods and tools, and a process. There is also an International Standards Organization (ISO) standard 15288 documenting and describing the underlying processes that typically make up a system life cycle.[18] This paper adopts the International Council of Systems Engineering (INCOSE) definition because it is widely recognized and includes relevant keywords such as stakeholder needs, requirements, verification and validation:[16]

> A transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods.

In *applied research*, the focus is often on transdisciplinary research.[1] For the projects studied in this paper, this research focused on a complex System of Systems (SoS). The definition of SoS adopted here:[19]

> A System-of-Systems is a collection of systems that maintain their operational and managerial independence.

To develop complex systems, we need people, processes, and supporting systems: also called a *sociotechnical* system or organization. A sociotechnical system can be described as a system that contains the subsystems *people* and *processes*, and the *methods*, *facilities*, and *equipment*, as shown in Figure 1. Organizational real-world systems are complex and messy[20] and cannot be analyzed in the same way as physical systems.[21–23]

Winter and Checkland[22] suggest the need for two viewpoints, *hard* and *soft*, each providing a contrasting image of managing projects. According to Crawford and Pollack,[24] there is some confusion between

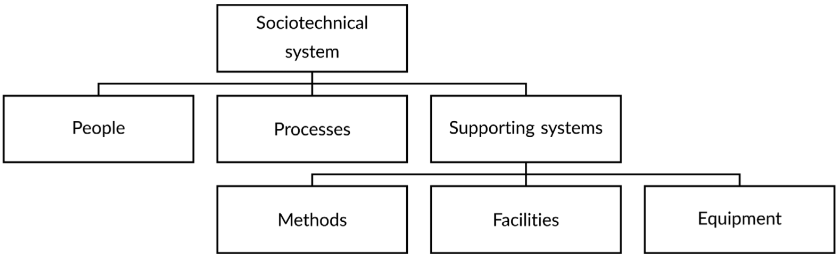


FIGURE 1 The sociotechnical system adapted from Pajarek[21]

the hard and soft paradigms in PM. The hard paradigm is philosophically grounded in positivism and realism, while the soft paradigm is grounded in interpretivism and the constructivist epistemology.[24] Furthermore, the hard paradigm is often associated with a linear approach to problem solving and management to realize the goal and manage the project life cycle. In contrast, in the soft paradigm viewpoint to project management, the manager is continuously observing and evaluating the situation, and can make a choice to take action to improve or change the project.[22] This does not mean that the hard paradigm cannot take an iterative approach to problem solving, but the goals and the approach to achieving the goals are known and planned. Hard systems thinking aspires to about "an efficient means to achieve a predefined and agreed end,[25]" while soft systems thinking methods are based on "interactive and participatory approaches to assist groups of diverse participants to alleviate a complex, problematic situation of common interest.[25]"

Some have claimed that SE employs "common sense" principles[26–28] and others suggest its value may be underestimated and its benefits underrepresented in the literature. There have been efforts,[29] most notably by Honour[12,30] and Boehm et al.[31] to quantitatively measure the benefits of SE activities in projects. Honour[12] found an optimal level of SE activities in a project of 15%–20% of the total effort. However, this was a limited study with self-reporting primarily by systems engineers and their individual perceptions. This study was continued and in Honour,[30] the aspects of technical quality and program success concerning SE activities were discussed.

Furthermore, a division of effort to the different SE activities of: *mission definition*, *requirements engineering*, *system architecting*, *system implementation*, *technical analysis*, *technical management*, *scope management* and *verification and validation* was suggested, where verification and validation clearly came out as the prime benefactors of SE activities.[30] Boehm et al.[31] looked at software projects and measured the Return on Investment of applying SE. They found a relationship between SE activities and software productivity, and that even minimal SE efforts would increase the project productivity significantly.[31] Cook and Wilson[27] describe which types of activities yield the most significant value in a project's life cycle, and how the advent of Model-Based Systems Engineering (MBSE) may bring added further value to projects and may be more easily linked to traditional engineering activities. They also touch on what is necessary to have a *good* SE environment, such as clear and shared objectives, a common model of system and worldview, an understanding of the process, and a stable environment and context in which the system is developed and deployed.[27] Even so, there are obstacles and barriers to introducing SE in any organization. Some organizations believe that using SE *processes* may hinder creativity because they associate the process with a prescriptive, detailed, flow-diagram approach which forces your work process.[32] Sheard et al.[32] continue to list other barriers such as: poor definition or understanding of SE, applying SE without a specific purpose, and lack of resources.

2.2 | Integration of project management and systems engineering

The systems developed in the past decades are more complex and require a higher level of coordinated engineering and management.[33] INCOSE, the Project Management Institute (PMI) and the Massachusetts Institute of Technology (MIT) established an alliance team in 2011 to analyze the integration of SE and PM, based on the recognition that these roles have overlapping and complementary responsibilities. Both are concerned with running a project and delivering a system that satisfies the needs of their stakeholders. Separately, INCOSE produces the SE Handbook which includes processes and guidelines for managing a system life cycle, and the PM Body of Knowledge (PMBOK) does the same for project management processes. We apply the PMBOK definition for Project Management:[34]

> Project management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements.

An early study from industry recognized that through their shared concern of meeting a customer's needs, SE and PM should be integrated both by functional decomposition and by practical integration[35] and suggests that teamwork is the key to making it successful. Roe[36] discusses the integration of PM and SE in an Integrated Product Development setting. Smith and van Gaasbeek[37] use the analogy of the DNA double helix to represent an inherent need for integration of the two for project success. Johnson[38] discusses the history of, and similarities and differences between, PM, SE, and Operations Research. Other research suggests that sharing a common language and understanding of responsibilities is necessary to make the integration work.[39–41] Xue et al.[42,43] describe a practical case of applying integrated SE and PM to a student engineering project. They introduced a framework to support the tailoring and application of such activities and showed its utility. Traditional PM has been criticized for restricting innovation and creativity[44,45] because of its strict processes require detailed planning and scoping before starting the project. For innovation and product development, it may not be possible to plan to the level of detail needed by these guidelines, and, may restrict the solution space alternatives. The terms *research* and *innovation* may be used interchangeably in product development, but the terms may have different implications depending on how the end goals are framed or expressed. The *agile* project management approach is grounded in enabling the ability to respond to changing circumstances and new discoveries. It also means empowering the whole project organization to participate in making decisions, instead of relying on the project manager to decide the scope and team activities. There will still be project management activities, but the top-down hierarchical chain-of-command is replaced, reducing some of the asymmetry between power and influence over work. Simultaneously, agile project management is not equal to the absence of management,[46] but rather a shared team leadership and management.[44]

However, it is rare to receive training in both SE and PM, as they are typically separated in academic or training environments. Cohen et al.[47] piloted a training simulator for systems engineers to learn PM in their graduate studies. They concluded that simulations help build the practical understanding[35] needed, and that further scenarios should be developed for training as well as comparing with real-life situations.[47] Baron and Daniel-Allegro[48] provide the results from an online course for improving systems thinking through embedded systems projects. This proved helpful for the students participating. The authors highlighted the outreach potential of online courses and the possibility for distance learning. Furthermore, training in PM relevant to research projects is lacking.[6] There is also a need for hands-on training and a good understanding of the field in which the projects are executed.

2.3 | Applied research projects

Academic research activities are increasingly project-based and applied.[1,6] Applied research projects are linked to *Mode 2* research which includes "collaborative and transdisciplinary research, greater heterogeneity in the sites of knowledge production, deeper social accountability and broader forms of quality control."[1, p. 690]

In comparison, basic research is focused on "advancing knowledge for its own sake."[1, p. 690]

An observation that motivated the research questions was of doctoral researchers who balance researching remote sensing and engineering tasks. When building a satellite, there are many engineering tasks such as circuit board layout, mechanical design, physical integration, vibration and shock testing — all of which are everyday in the industry. They do not necessarily yield research data for publication in remote sensing journals. However, since the satellite is the foundation for the remote sensing system, it needs to be engineered and built to deliver data. In this situation, the doctoral researcher must manage their tight schedule to deliver both engineering product and research results. At the same time, one can argue that the problem with this example is poor project planning, and that the university should have taken engineering tasks into account from the start of this research project. A more informed assessment during the application phase could anticipate these needs and plan for them. This suggests the importance of involving the broadest set of disciplines in the proposal writing and work-package definition phases.

Fowler, Lindahl and Skjöld[6] discuss the application of PM in universities based on an empirical study. Traditional PM was developed for the linear execution of pre-defined tasks/goals, countering the iterative research and knowledge-building trajectory. They found that with the projectification of research projects through funding mechanisms, researchers "indeed feel compelled to appropriate and use PM to become viable for funding.".[6, p. 11]

In the article, the interviewees discussed the concept of project start and end in the context of research. It is essentially a continuous effort that does not have a clear start and end, except within the context of individually funded projects or assignments. The authors also found that there is a separation between the project leader and the project manager, where PhD and post-doc candidates are often given the more practical and administrative tasks in a research project. Finally, they list barriers to implementing PM: (1) PM requirements in projects "have little relevance for how the research should in fact be carried out,"[6, p. 25] (2) projectifying the administration separately from the research work, and (3), division of labor between researchers and project administrators (who technically have the role of researcher but end up being responsible for the PM tasks.)

2.4 | SE and PM for applied research projects

The role of SE and PM in applied research projects is not well-defined. SE and PM practices are documented through their processes, such as in ISO standards, and these are commonly authored by industry practitioners. Research processes may not be communicated in the same way. Still, most researchers follow a simple workflow, as shown in Figure 2, and scholars can find guidelines[49,50] on how to run a research project and suggestions for qualitative or quantitative methods.[51]

However, given the definitions provided in the earlier sections and the description of SE and PM roles in the SEBoK[16] and PMBOK,[34] there are qualities that researchers could aspire to apply to managing their projects. For example, using data-driven decision-making; applying holistic thinking; defining lifecycle processes; project planning, monitoring, and controlling; demonstrate end-user awareness and stakeholder analysis; continuous development; teamwork; managing technical and project risk.

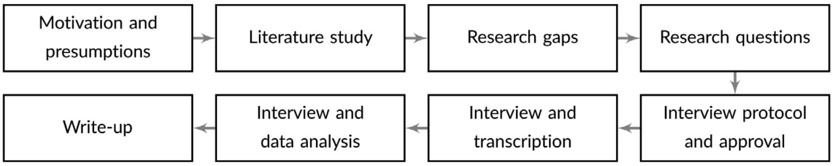


FIGURE 2 The research method applied

3 | RESEARCH METHODOLOGY

3.1 | Study method

The research is based on a qualitative case study, and follows a workflow shown in Figure 2. The data sources include a literature review to substantiate the knowledge gap and semi-structured interviews of between 45–60 minutes of 18 participants. The participants were selected based on their involvement in space projects in two university-based institutions: using a *key informant sampling* method.[52] The key informant sampling method was chosen to collect in-depth information relevant to the research questions, with a mix of people representing different roles within their respective projects. The informants perform research based on space knowledge and are also responsible for technology development, and practice applied research.[1] Half of the informants (9) are employed by a research institute, and the other half are members of various departments within the faculty of engineering at a public university. The informants all have a MSc degree, and most have PhD degrees. The organizations are in the same city and can be considered to be influenced by Scandinavian socio-cultural norms.

In the academic organization, there was an effort to introduce more SE both into the curriculum (by introducing systems engineering classes in 1st, 4th, and 5th year courses), and in faculty membership (hiring of one adjunct professor and one associate professor in SE). The results of these efforts were not clear at the time of the study, except that more people knew of the SE concept and had heard the term previously.

3.2 | Interview and data analysis

The interview protocol was based on the research question developed through an iterative process using relevant literature. A semi-structured interview format was chosen for a natural flow of a dialogue and allowed the interviewer to ask additional questions, if needed.[53] Introductory questions such as "Tell me a little about your background." or "What educational background do you have?" started each interview to help the informant relax and build rapport. The questions were posed in a combination of descriptive, ("How would you describe the research process?") and reflexive, ("What would you say are the benefits and challenges of systems engineering?") questions. All interviews were carried out face-to-face and recorded, and the interviewer took notes in case the recording was lost. A single researcher acted as the primary interviewer for all informants, while the second researcher listened and asked additional questions if needed at the end. The third researcher did not participate in the interviews, and only analyzed the transcriptions. The questions were available in two languages, and interviews were transcribed in their original language, either English or Norwegian. The informants were anonymized prior to analysis, and only the interviewer had the key to match the informant to a transcript.

An interview analysis protocol was based on Likert-scales of 1–5 (1 = to a low degree; 5 = to a high degree), with different statements the researcher would evaluate, given in Table 5. The interviews were analyzed independently by three researchers to provide triangulation on the results. The statements were based on the research questions, in addition to an evaluation of to which degree the informant had an engineering, educational, or research stance. The assessment of stance was based on how the informants identified themselves (for example if they said they were engineers, or if they said their primary role was as a lecturer), and what types of tasks they said they did in their jobs. The first round of analysis took place over the course of ten weeks, where the researchers independently evaluated the statements based on the interpretation of the transcripts. After that, the researchers met and discussed the results and explored the differences in rating where applicable. Finally, a score, S_x, was assigned to each statement based on the median of the researchers' scores. A median was chosen because it gives a measure of central tendency based on the rank of the score, appropriate for the non-continuous nature of Likert scales data. The Likert scale was further compacted to three levels: *low* for levels 1–2, *neutral* for level 3, and *positive* for levels 4–5 to enable more accessible discussion of results.

Spearman's ρ correlation coefficient was used to measure the relationships between Likert scale scores assigned to the fourteen protocol categories. The equation is given in Equation 1.

$$\rho = 1 - \frac{6\sum d^2}{n(n^2 - 1)} \tag{1}$$

where d is the difference between the ranks of the median Likert scores, and n is the number of questions. We also calculated the *p-value* for each ρ-value to signify statistical significance at the $\alpha = 0.05$ level. The coefficient measures the tendency for ranked values to change together. A so-called monotone relationship has a value between -1 and $+1$, where -1 is perfect negative monotonic, 0 is no monotone, and $+1$ is perfect positive monotonic. A perfect positive monotonic means that all data points in X increase as Y increases, and a perfect negative monotonic means X decreases as Y increases.

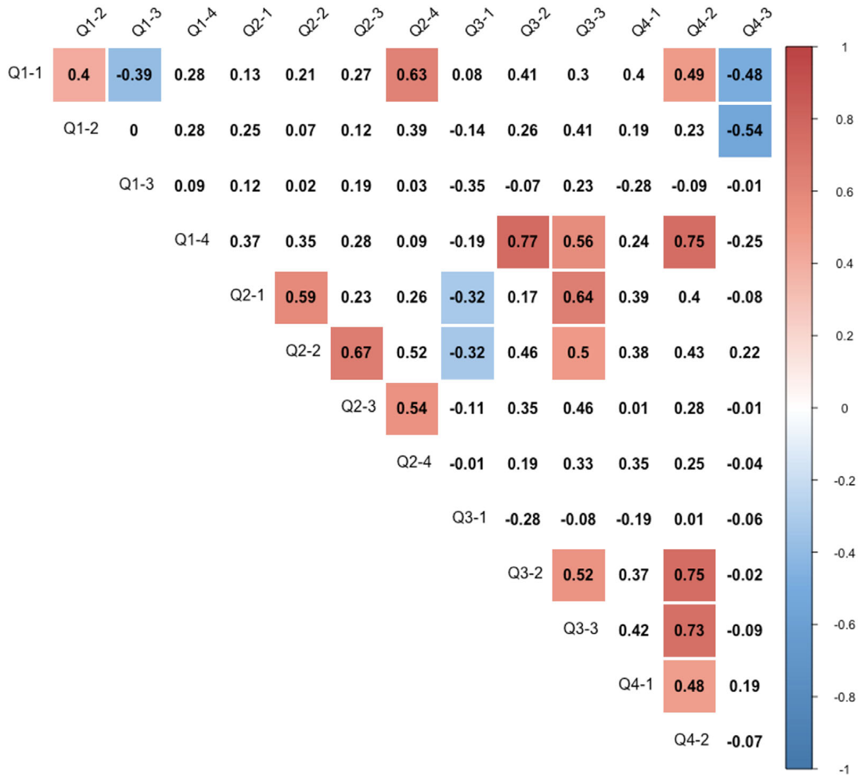


FIGURE 3 Correlation matrix of statements. The colored elements have a p-value larger than 0.05 (significant correlation)

3.3 | Validity and reliability

Lincoln and Guba[54,55] introduced four criteria for research trustworthiness commonly applied among social science researchers to sensitize reliability and validity to the specific nature of qualitative research: dependability, credibility, transferability and confirmability. The terms reliability and validity have by some been considered unsuitable for qualitative research[56,57] while others[58] use these terms but include several recommendations, including triangulation for enhancing quality. Triangulation is defined as "The use of more than one method or source of data in the study of a social phenomenon so that findings may be cross-checked."[57, p. 392]

The term also applies when multiple observers are employed to overcome the weakness or intrinsic biases and the problems that arise from a single observer, as done in this study.

The trustworthiness of the research presented here is ensured by addressing both the dependability (reliability), credibility (internal validity), transferability (external validity), and confirmability (construct validity) in the research design and data collection.[56] The triangulation strategy implemented improves the credibility and dependability of the case study research, i.e. data source (literature and a case study), data type (interviews and interview analysis), theory (PM and SE perspectives) and researcher triangulation of both theory and in the interview analysis. By addressing transferability similar to generalization, the study considers the extent to which the findings can be analytically generalized to other institutions or situations.

Confirmability refers to the extent to which others can confirm the findings, i.e. the reproducibility of the research. The confirmability of the study is obtained employing accurate and objective account of the concept under study, the research problem, case studies, research approach and the construct under investigation. However, given the inherent weaknesses of qualitative research methods and that most social settings are contextually unique, the authors acknowledge some limitations regarding construct validity. The impact of these limitations on the interpretation of results and conclusions are discussed in Section 5.

4 | RESEARCH FINDINGS

In this section we present the main findings from the interview analysis and quotes from the interviews that address the research questions.

TABLE 1 Statements used to guide the researcher assessment of interviews. Q1-Q4 were evaluated on a Likert scale

ID	Topic	Relevant to RQ no.
Q1	Understand the academic stance of the Informant	
Q1-1	To what extent does the Informant hold a research stance?	
Q1-2	To what extent does the Informant hold an educational stance?	
Q1-3	To what extent does the Informant hold an engineering stance?	
Q1-4	To what extent does the Informant understand systems engineering?	RQ3
Q2	Understand the stance/definition/explanations of project, process, task, and goals. Understand how the Informant balances between processes and goals	
Q2-1	To what extent does the Informant distinguish between engineering project and research project?	RQ1,RQ2
Q2-2	To what extent does the Informant distinguish between the engineering process and research process?	RQ1,RQ2
Q2-3	To what extent does the Informant distinguish between engineering tasks and research tasks?	RQ1,RQ2
Q2-4	To what extent does the Informant distinguish between research goals and engineering goals?	RQ1,RQ2
Q3	Understand if systems engineering could contribute towards research processes and goals	
Q3-1	To what extent does the Informant believe that SE is integrated in academia?	RQ3
Q3-2	To what extent does the Informant believe that SE should be integrated in academia?	RQ3
Q3-3	To what extent does the Informant believe that SE could be integrated in academia?	RQ3
Q4	Understand the stance on different types of management	
Q4-1	To what extent does the Informant distinguish between research and engineering management?	RQ2
Q4-2	To what extent does the Informant distinguish between research and project management?	RQ2
Q4-3	To what extent does the Informant distinguish between project and engineering management?	RQ2
Open-answer questions for the analysis		
Q5	What are the greatest benefits of systems engineering?	RQ3
Q6	What are the most challenging aspects of systems engineering?	RQ3
Q7	What, if anything, separates an engineering project from a research project?	RQ1,RQ2
Q8	What, if anything, would be the benefits of more knowledge/support to project and engineering processes in academia?	RQ1
Q9	To what degree did the SE course influence the Informant? What thoughts does the Informant have about the course?	

Notes: Q5-Q9 were open-ended questions that were evaluated based on overall impression and direct quotes from the interview transcripts.

The quotes have been translated from Norwegian to English by the authors. The correlation coefficient results of the interview analysis are given in Figure 3. The relationships which have a lower *p*-value than 0.05 are highlighted in the matrix with a background color. Some correlate with the literature but were not directly analyzed by the statements in Table 1. The main statistical correlations were: (1) Opinion on the integration of SE in academia is linked to the understanding of SE; (2) There is a perceived distinction between research and project management; and (3) The variety in goals and tasks distinguish research and engineering.

4.1 | Tabulated results

Tables 2, 3, and 4 list the data from the interview analysis on Table 1 questions, the counted informants per Likert level (1–5), and compacted counted informants per categorized (low–neutral–positive) Likert level.

Infromants may have a combination of stances (Q1-1 to Q1-3), and 13 of 18 had a research stance, 5 of 18 had an educational stance, 11 of 18 had an engineering stance, see Figure 4 for a mapping of stances.

4.2 | Main findings

4.2.1 | Opinion on integration of SE in academia is linked to understanding of SE

All informants were asked to state their understanding of systems engineering and were encouraged to reflect on what it meant for them and whether it had a place in research and academia. A typical answer that resulted in a high score is when many of the keywords given in Section 2 are included.

> "SE is everything related to systems. From you have an idea until you have an existing technology and you need

TABLE 2 Median score results of interview analysis based on Likert scale of 1–5

Informant no.	1	2	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Q1-1	1	5	4	4	4	3	2	4	2	4	5	5	5	5	4	1	5	5
Q1-2	2	3	4	1	1	1	1	1	3	3	5	3	2	4	4	3	5	2
Q1-3	5	2	5	2	2	5	4	2	4	4	1	4	3	4	2	4	4	4
Q1-4	3	2	4	4	4	4	2	1	3	4	4	2	4	5	4	4	4	5
Q2-1	4	4	4	4	2	4	4	3	3	4	5	2	4	4	3	4	4	5
Q2-2	4	4	4	5	3	4	4	3	3	5	4	3	4	4	4	2	4	4
Q2-3	4	4	4	4	3	3	4	3	3	4	2	3	4	5	4	2	4	4
Q2-4	4	5	4	4	3	3	4	3	3	5	4	5	4	4	4	3	5	5
Q3-1	2	2	1	2	2	1	2	3	3	2	2	2	2	2	2	2	2	3
Q3-2	3	3	4	4	5	4	3	3	3	5	4	3	4	5	4	3	5	4
Q3-3	4	3	4	3	3	3	3	3	3	4	4	2	4	4	3	3	4	4
Q4-1	3	4	5	4	4	3	3	3	3	4	5	3	4	2	3	3	4	4
Q4-2	3	2	4	4	4	3	3	3	3	5	5	3	5	5	3	3	4	5
Q4-3	5	3	3	4	5	3	4	3	3	4	3	3	3	1	3	3	3	3

TABLE 3 Counted tabulated results of interview analysis based on Likert scale

Likert score	Q1-1	Q1-2	Q1-3	Q1-4	Q2-1	Q2-2	Q2-3	Q2-4	Q3-1	Q3-2	Q3-3	Q4-1	Q4-2	Q4-3
1	2	5	1	1	0	0	0	0	2	0	0	0	0	1
2	2	3	5	3	2	1	2	0	13	0	1	1	1	0
3	1	5	1	2	3	4	5	5	3	7	9	8	8	12
4	6	3	8	10	12	11	10	8	0	7	8	7	4	3
5	7	2	3	2	1	2	1	5	0	4	0	2	5	2

> to connect to it. You need to write and specify this system. You need to write requirements' definitions. (…) And you have different types of diagrams, context diagrams, class diagrams…that you use to describe your system. And then you need to be able to document it. And plan how to test and verify and validate it. And you need to understand the regulations. And quality systems. You need to have an understanding of electronics, mechanics, how things are linked together. (…) You need to understand the context of what you're working on [Informant 15]."

Understanding of SE (Q1-4) was positively correlated with (Q3-2) believing that SE should be integrated in academia ($\rho = 0.77$, $p = 0.00021$), and with (Q3-3) believing it could be integrated ($\rho = 0.56$, $p = 0.015$). Having an understanding of SE (Q1-4) also correlated positively with (Q4-2) differentiating between research and project management ($\rho = 0.74, p = 0.00037$). According to Table 3, only four informants were scored low or neutral on the understanding of SE (Q1-4), shown in Figure 5. We found no relationship between stance (Q1-1, Q1-2, Q1-3) and understanding of SE (Q1-4).

The informants noted that to them, SE was common sense and recognized the SE processes from how they already worked. Furthermore, that having implicit knowledge and applying "common sense" processes and principles may be challenging when there is a personnel turnover.

> "I feel like we have it already. We do not talk about it in the formal way, we just do as we always have done. At a certain level I feel like we have those processes

TABLE 4 Compacted counted tabulated results of interview analysis based on categorized Likert scale

Likert score	Q1-1	Q1-2	Q1-3	Q1-4	Q2-1	Q2-2	Q2-3	Q2-4	Q3-1	Q3-2	Q3-3	Q4-1	Q4-2	Q4-3
1–2 (not)	4	8	6	4	2	1	2	0	15	0	1	1	1	1
3 (neutral)	1	5	1	2	3	4	5	5	3	7	9	8	8	12
4–5 (positive)	13	5	11	12	13	13	11	13	0	11	8	9	9	5

TABLE 5 Comparison of the linear, agile, and research development environment

	From Paluch et al.[62]		Research findings	
Characteristics	**Linear development**	**Agile development**	**Applied research**	**Engineering**
Solution space	Solution space defined.	Solution space undefined.	Solution space undefined. Theoretical grounding.	Solution space and boundary conditions defined.
Customer	Stable and known customer preferences. Limited customer willingness to interact. Customer in need of fully specified product.	Changing and/or unknown customer preferences. High customer willingness to interact. Customer open to engage with interim products.	Changing research community interests. High willingness to review. Research community open to contribute to products.	End-users are typically known. Stable and known preferences.
Task	Low task modularity.	High task modularity.	High individual research modularity. Managing researchers and research projects. Applying for research funding. Explore deep into the topic and develop knowledge.	Implementing the details. Integrating systems. Make a product quickly.
Goal	Well-defined and agreed-upon goals.	Open and agreed upon goals.	Open goals, not well-defined. Creating knowledge for a better world. Answering research questions.	Known, defined goals. Solving a problem. Product delivery.
Process	Well-defined and standardized process.	Adaptive process models. Continuous integration, test-driven.	Weakly defined, but highly adaptive process. Highly learning-focused process. Known and declared methods in some fields.	Strict processes with higher maturity projects. Less strict with low-maturity projects.
Organizational	Low tolerance for interim failure. Strong need for managerial control.	High tolerance for interim failure. Weak need for managerial control. Shared ownership.	Weak need for managerial control. High tolerance for interim failure. High acceptance for individuality.	Need control in large companies, not necessary for smaller.

Notes: The linear and agile development columns are from Paluch et al.[62] while the Applied research and engineering columns are based on the findings of the case studies.

already. Without thinking over what it means or explicitly describing them. (…) That might be the challenge today, to onboard (…) and transfer the knowledge. We who have worked a couple of years have lots of implicit knowledge, which we don't think about in the daily work, but just do. New people need to be trained in the routines and the way of working. They may have similar experience from other employers, so they are probably not strangers to the way of working. But maybe not with the same vocabulary [Informant 17]."

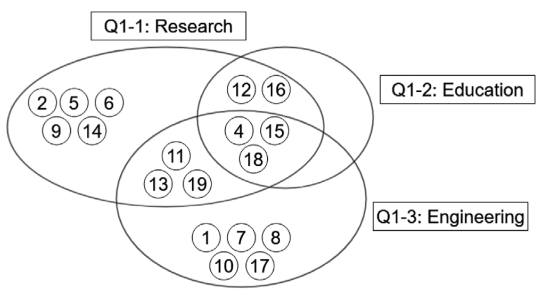


FIGURE 4 Venn diagram of informants and stances

4.2.2 | Distinction between research and project management

There was a significant positive correlation between the informants (Q1-1) holding a research stance and (Q4-2) differentiating between research and project management ($\rho = 0.49$, $p = 0.0040$). However, there was a significant negative correlation between the informants (Q1-1) holding a research stance and (Q4-3) differentiating between project and engineering management ($\rho = -0.48$, $p = 0.0043$). Similarly, there was a significant negative correlation between the informants (Q1-2) holding an educational

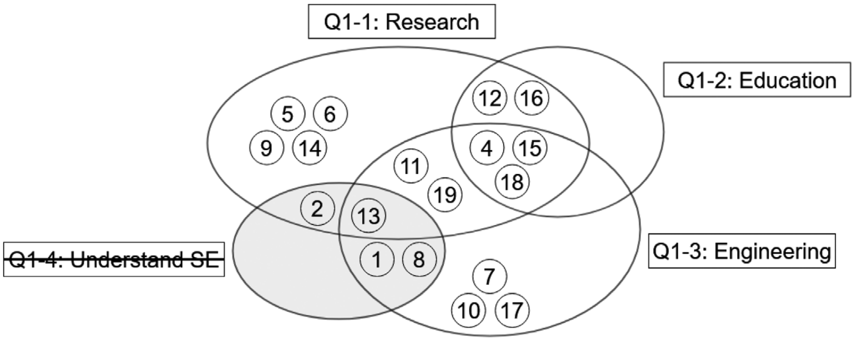


FIGURE 5 Venn diagram of informants' stances, and indicating which were perceived to not understand SE (gray circle)

stance and (Q4-3) differentiating between project and engineering management ($\rho = -0.54$, $p = 0.0022$). Some informants explained engineering management in the same way they described project management, with a clear end-goal and schedules and boundary conditions. The informants' stated that in practice, in academia, engineering projects are run more like projects traditionally are run, where the scope and requirements are agreed upon beforehand.

Three types of projects in academia were described by some of the informants: the education of doctoral students, the actual research projects from when funding arrives until the end of funding, and the continuous project of determining how to push and contribute to the research front. These projects have different time scales and needs, and should be managed differently. Additionally, each instance of these projects needs tailoring to support the specific project objectives. For example, two doctoral students may have very different plans in terms of laboratory equipment and experiments, or active supervision needs. However, both doctoral students have to complete coursework, submit research articles, and defend their thesis on time, and as such the project follows a pattern with known objectives.

4.2.3 | The goals and tasks distinguish research and engineering

The informants were asked to compare research and engineering in terms of tasks, projects, goals, and processes. There were positive correlations between the informants who differentiated between (Q2-4) research and (Q2-2) engineering goals with engineering and research processes ($\rho = 0.52$, $p = 0.027$), and with (Q2-3) engineering and research tasks ($\rho = 0.54$, $p = 0.022$). There was a positive correlation between the informants holding a (Q1-1) research stance and (Q2-4) differentiating between research and engineering goals ($\rho = 0.63$, $p = 0.005348$).

The informants were asked to describe the engineering and research process. Many of the informants related the process to the end goal and end product, and did not elaborate on the procedure or process used. Most of the informants stated that there were not clear guidelines for processes or methods for managing projects at the university.

Researchers have been asked to do engineering tasks, which may or may not contribute to publishable research results. There were different opinions on whether or not researchers should do these in order to produce answers to the research questions. On the one hand, to fully understand the measurements you are producing, you should fully understand the instruments that provide the measurements. On the other hand, if all researchers should understand their instruments or infrastructure fully, it would take too long to push the research frontier.

> "When you say research project I think basic research. While when you say engineering I think something needs to be developed because there is a specific task, you are developing equipment for a function. Both use a scientific approach. You have a hypothesis which needs testing, and you evaluate the results in the end [Informant 17]."

> "If you as a physicist are doing an experiment where you need electronics. You are doing the research. While the person making the electronics is just making the electronics. Pushing the boundaries for electronics for making an instrument, I don't view that as research. (…) Incremental development, which instruments exist or what they measure or which measurement electronics are incremental research. I am hesitant to say that it's research [Informant 1]."

> "To me, if you cannot build your infrastructure you cannot use it. I don't consider it engineering. It is a natural part of being a researcher. (…) For maximum performance you have to know everything about your lab and you can only do that by being hands-on. (…) Standard maintenance is a part of the social research context. I actually consider that all doctoral researchers shall contribute with some sort of technology for the lab [Informant 12]."

4.3 | Other findings

4.3.1 | Perceptions on management

In terms of what constitutes good management, although not asked explicitly, some of the informants discussed positive experiences of management and their experiences with management at the university. Informants also described how "too stiff" or "too strict" management could hinder the progress.

> "Iterative discussions. Have short and frequent meetings, try not to plan too large increments between the milestones — make them smaller to give people a chance to come up with ideas and concepts. Evaluate them strictly but keep the ideas coming and in the knowledge base. (…) So I think you could call it lean. That you try to develop things as time moves ahead, and that you're not too rigid with the specification [Informant 4]."

> "I believe in the cooperation between the technical system manager with the project manager. (…) If you can get a good symbiosis between those who can run and manage the project, and those who works with and understand the architecture well. (…) Good people chemistry and workflow. You need dynamic people in both roles who don't care too much about rigidity and hierarchy. (…) You need to be relaxed and with the mindset of helping each other. Of course you need specific role definitions as you move up in the organization, but I think if it gets too stiff it stops working. I also think it has a lot to do with personalities. [Informant 4]"

There was an agreement that universities were not structured for running projects as viewed from an engineering perspective. However, there are administrative support functions, and guidelines and support for preparing funding applications.

> "There has been little support from the university for the execution of the project. We have gotten rooms and areas, and the scientific employees are available to answer questions. But you have to figure things out yourself. (…) There is no template for how you run a project. But you might not want one to exist either [Informant 1]."

4.3.2 | Processes can hinder creativity and require resources

The informants were asked to discuss what they saw as challenges or negative sides of having SE or PM processes in research. Some of the informants mentioned how strict processes might hinder creativity. Another challenge was that implementing processes and training people is costly and may be challenging to show the Return on Investment of such efforts.

> "It [SE] should not be a straitjacket that limits the craziness in your ideas. But some understanding of how the world works is good [Informant 15]."

> "For small companies and small projects it is not possible to apply the full systems engineering process. The question is how to find the right balance. How do we develop or find the tools that suit our processes and capabilities. I think that is the work that needs to be done. (…) We would definitely benefit from the systems engineering way of thinking in the Research & Development and European Space Agency (ESA) projects [Informant 6]."

4.3.3 | SE gives a holistic overview and structure to applied research projects

In the interviews, the informants were asked to reflect upon what they saw as benefits to having more SE in their research environment. A repeating theme revolved around having enough people and resources to do the activities and enough knowledgeable people to do them. For the projects with explicit resources allocated to either PM or SE, informants saw clear benefits. For the projects with no explicit resources, but with thoughts of using the practices, it was difficult to see a clear connection between resource use and utility. Finally, sometimes the projects have just the right people with the knowledge to introduce and use the practices in the right way.

> "Everyone working in research that is to be applied, infrastructure, platforms, can benefit from [SE] [Informant 15]."

> "Holistic thinking puts things in perspective. Within signal processing, we try to make something *epsilon better*. In a communication system we have many algorithms in a pipeline, each epsilon better — but the system may not be better as a whole because the epsilons cancel each other out. And then you've written 5 articles about something that doesn't help anything, except feeding academics [Informant 18]."

> "I feel like systems engineering is a good tool to get a holistic picture and ensure that things flow together and that everyone contributes, or at least that they can

contribute. In the right order. Piece together the good puzzle it should be in the end [Informant 17]."

5 | DISCUSSION

During the course of this case study it became clear that with the increased challenges of complex systems, there is a need to enable researchers to manage applied research projects concurrently with other university duties. There is not a clear distinction between research and engineering tasks, and it is challenging for researchers to have the resources and time needed to manage the complex technical infrastructure requiring to perform research activities. The perception of academics on SE and PM for applied research projects, and the distinction between research and engineering, are the key contributions of this paper.

5.1 | Implications

The goals of the work performed in many cases may be used to distinguish between engineering and applied research,[59] while keeping in mind that the source of the goals (individual scientist? Society? End-user?) matters. According to Niiniluoto,[59] "the knowledge provided by applied science is expected to have *instrumental value* for the associated human activity."[59, p. 6]

However, Niiniluoto also argues that a practitioner of applied science (or research) using knowledge gained to solve a problem, is not doing science. Stuart[60] suggests several methods for different areas of applied research to provide the needed end product, from the viewpoint of engineering design, and comments on how research in applied science can contribute to satisfying societal needs. In the field of artificial intelligence, early work was published through application studies, which "could uncover deficiencies in the current body of scientific knowledge."[61, p. 128]

However, as the field has matured, topics published 20 years ago could be considered engineering today, and not research because it is practiced in industrial settings.

Some of the informants' impressions support taking a soft and sociotechnical viewpoint to management analysis in academia.[21,22] Supervising researchers (such as PhD or PostDoc) requires interpersonal skills, and needs tailoring depending on the specific research project. The informants highlighted flexibility, and agility, confirming some of the literature on good management.[8,22,46] Managing applied research projects also requires a soft approach,[22] based on the complexity of the project that includes multidisciplinary people and components. The research process can be considered a "messy real-life situation," which cannot be planned and detailed ahead of time, and the goals (apart from contributing to the knowledge frontier) keep changing.

The study presented in this paper contributes to the discourse on the demarcation between engineering and applied research, by offering different perspectives from academics mainly in engineering fields, who largely do applied research projects[1] that need engineering support. Furthermore, this study can inform on the application of systems engineering and project management practices in applied research projects. The results confirm some of the findings from Malik et al.[44] as the informants were hesitant to apply too much process because it could hinder the creative flow of research. Furthermore, there is a strong agreement that much of SE is "common sense," but that it helps sharing a common language to enable collaboration.[26–28]

There is a difference between the formalization of projects and executing project activities. The increased complexity drives the need for planning and management of research projects, and if the researchers are not trained they will choose their methods and tools arbitrarily. Researchers will execute their research projects and deliver research results even without formalized project activities, as they have been doing until now.

In some cases, there will be little distinction between the planning of and execution of research projects. The process of writing research funding applications may include some high-level planning, but in practice does not include *how* the project can be implemented. It is not surprising that a good understanding of SE correlated with informants' belief that SE should be integrated in academia, because most peoples' understanding of SE is based on their personal interest in SE as it is not explicitly taught in general courses. A possibility would be for the university to offer short courses or training in systems thinking and short introductions to relevant SE and PM skills. This could be done in collaboration with other universities, to lower the cost for setting it up. Baron and Daniel-Allegro[48] give an example of a successful MOOC (massive open online course) and its success in developing systems thinking skills.

Research findings are usually the output of a research project, analogous to the "product" that the industry delivers, where the research community is comparable to paying customers. Researchers are continuously asked to publish their results (product), during which they go through a peer-review process (analogous to prototype feedback from end-users and managing risk), are asked to make improvements (analogous to iterating on design), and publish (analogous to introducing a product in the market place). Once the research is published, other researchers may build on that knowledge to create new knowledge. The original research group may continue developing additional knowledge, circling with an iterative product delivery where the product delivery responds to the customer feedback.[46] Another feature of research observes that while individual researchers focus on a specific small part of the field, together researchers form a community that encourages feedback and where "failures [are] not considered as defeats but as valuable opportunities for learning."[62, p. 499]

Many of the informants commented that it had become more challenging to do research without being supported by engineered infrastructure. Furthermore, that engineering support alone was not sufficient for the researchers, but that the community needed research engineers — typically people with higher-level research experience and engineering know-how, to maintain and innovate the infrastructure and labs. However, the way research projects are funded does not always allow for this, or the departments do not have adequate resources.

Furthermore, it is not clear what role SE and PM have in academic research projects. The data shows agreement that holistic thinking is valuable, and this suggests that researchers should have this skill. It could also be helpful for managing stakeholder expectations, and to plan and monitor research projects to help departments manage their engineering infrastructure better. SE and PM offer project performance measurements,[63] and these could support the management of applied research projects and engineering infrastructure. None of the informants mentioned applying analytics or measurements to their research projects, nor were they specifically questioned about this. There are promising opportunities for including AI and ML for PM activities and performance measurements,[64] which could be applied for research institutions. Still, there is a need for training of personnel and adoption of new techniques and systems.

This study supports the theory of the research organization as a sociotechnical system, as shown in Figure 1. There are different perceptions of the meaning of a research project, process, task, and systems engineering. Furthermore, there are different approaches to leadership, and the interviews highlighted that a tailored approach is needed for different research projects and at different levels. A challenge for researchers today is how to separate and balance their time between research, engineering, education, and project management. Although challenging, a practical implication from this study is that projects to a greater degree separate the engineering and research tasks assigned to researchers, to enable the researcher to focus on value-adding activities. This could also, over time, allow for better allocation of engineering resources in the department.

5.2 | Addressing the research questions

This paper laid the results of a study on how systems engineering and project management could benefit academic research projects by analyzing through the analysis of two case studies based on project data, participatory action research, and 18 semi-structured interviews. The findings from this study can improve the way research projects are managed in academia, by addressing the research questions:

- **RQ-1**: How can an engineering project ensure the fulfillment of academic research goals (in a university setting)?
- **RQ-2**: How can engineering goals and individual research goals be fulfilled simultaneously?
- **RQ-3**: How do researchers understand PM and SE?

For RQ-1, there were different opinions on what constitutes research and engineering. Most research topics today involving technology need engineering to push the research boundaries. Either because one needs engineering work to build scientific equipment, or the scientific research in developing technology needs to be integrated and tested. Engineering and research are more and more intrinsically linked in the applied research domain.

In addressing RQ-2, the concern is balancing the workload for research and engineering. There was agreement that in applied research the projects today are so complex that the traditional PM heuristics fail to support the process. Engineering tries to plan all activities and all requirements to meet specific goals, while research by nature is more iterative while moving towards a desired end-state. Perhaps what is needed is a more robust and systemic approach to research, and guidelines to enable researchers to distinguish between engineering and research. If one acknowledges incremental engineering as a part of the research, the gap between engineering and research engineering narrows.

For RQ-3, we found no clear data on how well researchers understand either SE or PM. However, people with a research stance do not distinguish between project and engineering management. People with a research stance differentiate between research and project management. Table 5 summarizes the distinction between engineering and applied research tasks, processes, projects, and goals based on the interview analysis and the literature in which the linear life cycle development and agile development were compared.[62]

5.3 | Limitations

The disadvantages of using interviews as a data source include sampling bias, interviewer bias, and interviewee bias. A potential challenge was that the interviewer could be considered an expert in the field compared to the interviewee. While this, on the one hand, allows for more straightforward exploration of topics and knowledge to carry informed conversations, on the other hand, this can also make the interviewees insecure, such that they try to "perform" to prove that they also know the topic because they feel the interviewer is testing them.[53] Because this study is interested in understanding how researchers view PM and SE, it was considered a strength that the interviewer was knowledgeable in the field to be able to follow up interesting topics in the semi-structured interview, which a less familiar interviewer may overlook. One researcher carried out all the interviews, while the second researcher listened and contributed with additional questions at the end of an interview. Both interviewers had knowledge and experience in PM and SE. The third researcher asked to analyze the transcriptions did not have strong knowledge of SE, but was given the definition from the INCOSE to compare against as a reference. The interviewer mentioned that it took some time to get accustomed to performing interviews, and that the quality could have been improved by training. If interviews are chosen as a data source for future studies, the authors recommend applying a pilot interviewing phase to improve the interview guide and "train" the interviewer. The key informant sampling method applied is helpful at the beginning of a study such as this, but is limited and cannot support generalizations.

6 | CONCLUSION

Autonomy of the researcher is a principle that "goes against" conventional PM. In traditional PM, the manager, be it the systems engineering manager or project manager, decides the scope of work and how and when it should be performed and completed.[44] For some researchers, they may feel that these restrictions hinder creativity and limit the research process of exploring new avenues and theories that were not pre-defined in the project.

Future work and research ideas:

- *Are research funds managed efficiently?* Research projects have goals for publications, for graduating PhDs, dissemination, and impacts. They often include a set schedule and limited funding. A future study could investigate which indicators are relevant to track for applied research projects, the historical track record for meeting goals, schedule, and budget constraints. The new study could categorize the findings according to type of project, and fill a gap in the literature by expanding on the relevance of this study.
- *Should research projects include funding for engineering tasks?* We found that for applied research, there is a need for resources for engineering tasks and infrastructure. A broader study into which types of research projects need this additional support would enable departments to prioritize resources during the proposal writing phase and researchers to be assured of this support so they may focus their time on research.
- *How do you measure the effectiveness of a research project?* Finally, what are the metrics or methods to measure the effectiveness of an applied research project to determine if it has been effective or not. There may be effects that will not materialize in the short run, but the project will still be effective.

ACKNOWLEDGEMENTS

This work was supported by the Research Council of Norway, Statoil, DNV GL and Sintef through the Centers of Excellence funding scheme, Grant 223254 - Center for Autonomous Marine Operations and Systems (AMOS) and the Research Council of Norway through the IKT-PLUSS programme grant 270959 (MASSIVE). It has been approved by the Norwegian Centre for Research Data (project no. 560218).

DATA AVAILABILITY STATEMENT

The data are not publicly available due to privacy or ethical restrictions.

ORCID

Evelyn Honoré-Livermore https://orcid.org/0000-0002-5664-330X
Knut Robert Fossum https://orcid.org/0000-0002-1020-730X
Erik Veitch https://orcid.org/0000-0001-6049-8136

REFERENCES

1. Bentley PJ, Gulbrandsen M, Kyvik S. The relationship between basic and applied research in universities. *Higher Education*. 2015;70(4):689–709. http://dx.doi.org/10.1007/s10734-015-9861-2
2. Tunstel E, Cobo MJ, Herrera-Viedma E, Rudas IJ, Filev D, Trajkovic L, Chen CLP, Pedrycz W, Smith MH, Kozma R. Systems Science and Engineering Research in the Context of Systems, Man, and Cybernetics: Recollection, Trends, and Future Directions. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2021;51(1):5–21. http://dx.doi.org/10.1109/tsmc.2020.3043192
3. Google. Outreach – Google Research https://research.google/outreach/ 2020 Accessed on September 04 2020.
4. Lenfle S, Loch C. Lost Roots: How Project Management Came to Emphasize Control over Flexibility and Novelty. *California Management Review*. 2010;53(1):32–55. http://dx.doi.org/10.1525/cmr.2010.53.1.32
5. Lenfle S. Floating in Space? On the Strangeness of Exploratory Projects. *Project Management Journal*. 2016;47(2):47–61. http://dx.doi.org/10.1002/pmj.21584
6. Fowler N, Lindahl M, Sköld D. The projectification of university research. *International Journal of Managing Projects in Business*. 2015;8(1):9–32. http://dx.doi.org/10.1108/ijmpb-10-2013-0059
7. Sharon A, de Weck OL, Dori D. Project management vs. systems engineering management: A practitioners' view on integrating the project and product domains. *Systems Engineering*. 2011;14(4):427–440. http://dx.doi.org/10.1002/sys.20187
8. Conforto E, Rossi M, Rebentisch E, Oehmen J & Pacenza M Survey Report Improving Integration of Program Management and Systems Engineering: Results of a Joint Survey by PMI and INCOSE tech. rep. 2013.
9. Kordova S, Katz E, Frank M. Managing development projects. –The partnership between project managers and systems engineers. *Syst Eng*. 2019;22:227–242.
10. Hamann RJ, Zandbergen BTC, Zijdemans PJ. Does Systems engineering in space projects pay? *INCOSE Int Symp*. 2009;19:1349–1361.
11. Honour E, Mar B. 9.2.8 Value of Systems Engineering - SECOE Research Project Progress Report. *INCOSE International Symposium*. 2002;12(1):450–455. http://dx.doi.org/10.1002/j.2334-5837.2002.tb02495.x
12. Honour EC. 6.2.3 Understanding the Value of Systems Engineering. *INCOSE International Symposium*. 2004;14(1):1207–1222. http://dx.doi.org/10.1002/j.2334-5837.2004.tb00567.x
13. Department of Defense Systems Engineering Management College. *Systems Engineering Fundamentals*. Fort Belvoir, VA: Defense Acquisition University Press; 2001.
14. Lippe S, Brocke J. Situational project management for collaborative research projects. *Proj Manage J*. 2016;47:76–96.
15. Rebentisch E, Prusak L. *Integrating Program Management and Systems Engineering: Methods, Tools, and Organizational Systems for Improving Performance*. John Wiley and Sons; 2017.
16. SEBoK (Editorial Board), RJ Cloutier (Editor in Chief). Overview of the Systems Engineering Body of Knowledge (SEBoK). https://www.sebokwiki.org/w/index.php?title=Systems_Engineering_Overview&oldid=61847 Retrieved 2020-09-21 from https://www.sebokwiki.org/w/index.php?title=Systems_Engineering_Overview&oldid=61847.
17. Johnson SB. *The Secret of Apollo: Systems Management in American and European Space Programs*. Johns Hopkins University Press; 2006.
18. International Organization for Standardization. ISO15288:2015(E): Systems and Software Engineering - System Life Cycle Processes Standard. Geneva, Switzerland; 2015.
19. Maier MW. Architecting principles for systems-of-systems. *Syst Eng*. 1998;1:267–284.
20. Yeo KT. Systems thinking and project management – time to reunite. *Int J Proj*. 1993;11:111–117.
21. Pajarek L. Processes and organizations as systems: when the processors are people, not pentiums. *Syst Eng*. 2000;3:103–111.
22. Winter M, Checkland P. Soft systems: a fresh perspective for project management. *Proc Inst Civ Eng*. 2003;156:187–192.
23. Donaldson W. In Praise of the "Ologies" A discussion of and framework for using soft skills to sense and influence emergent behaviors in sociotechnical systems. *Syst Eng*. 2017;20:467–478.
24. Crawford L, Pollack J. Hard and soft projects: a framework for analysis. *Int J Proj*. 2004;22:645–653.

25. SEBoK (Authors), RJ Cloutier (Editor in Chief). System Approaches, https://www.sebokwiki.org/w/index.php?title=Systems_Approaches&oldid=58931
26. Brill JH. Systems engineering– A retrospective view. *Syst Eng*. 1998;1:258–266.
27. Cook SC, Wilson SA. The enduring path to system success: investment in quality early-phase systems engineering. *INCOSE Int Symp*. 2019;29:817–835.
28. Vanek F, Jackson P, Grzybowski R. Systems engineering metrics and applications in product development: a critical literature review and agenda for further research. *Syst Eng*. 2008;11:107–124.
29. Carnegie-Mellon Univ. Pittsburgh PA Software Engineering Inst., Elm JP, Goldenson DR. The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey Tech. Rep. ADA610517 2012.
30. Honour E. 11.4.2 Systems engineering return on investment. *INCOSE Int Symp*. 2010;20:1422–1439.
31. Boehm B, Valerdi R, Honour E. The ROI of systems engineering: some quantitative results for software-intensive systems. *Syst Eng*. 2008;11:221–234.
32. Sheard SA, Lykins H, Armstrong JR. Overcoming barriers to systems engineering process improvement. *Syst Eng*. 2000;3:59–67.
33. Sauser B, Boardman J, Gorod A. System of systems management. In: Jamshidi M. Dr. ed. *System of Systems Engineering: Innovations for the 21st Century*. John Wiley and Sons, Inc.; 2008.
34. Project Management Institute. *A Guide to the Project Management Body of Knowledge*. 5th ed. PA, USA: Project Management Institute; 2013.
35. Lilburn Bert. Integrating Systems Engineering and Program Management. *INCOSE International Symposium*. 1996;6(1):689–696. http://dx.doi.org/10.1002/j.2334-5837.1996.tb02071.x
36. Roe Charles L.. Project management and systems engineering in an IPD environment. *INCOSE International Symposium*. 1996;6(1):1063–1068. http://dx.doi.org/10.1002/j.2334-5837.1996.tb02123.x
37. Smith Byron, van Gaasbeek Jim. Project Management, Systems Engineering and the Amazing DNA Double Helix. *INCOSE International Symposium*. 1996;6(1):882–886. http://dx.doi.org/10.1002/j.2334-5837.1996.tb02098.x
38. Johnson SB. Three approaches to big technology: operations research, systems engineering, and project management. *Technol Cult*. 1997;38:891–919.
39. Mooz H, Forsberg K. Visualizing system engineering and project management as an integrated process. In: *INCOSE International Symposium*. Vol. 7. Los Angeles, CA; 1997:573–580.
40. Emes M, Smith A, Cowper D. Confronting an identity crisis–How to "brand" systems engineering. *Syst Eng*. 2005;8:164–186.
41. Sharon A, Perelman V, Dori D. A project-product lifecycle management approach for improved systems engineering practices. *INCOSE Int Symp*. 2008:18.
42. Xue R, Baron C, Esteban P, Yang JB, Zheng L. Toward an improved monitoring of engineering projects. *IEEE Trans Syst Man Cybern Syst*. 2018;50:3541–3553.
43. Xue R, Baron C, Vingerhoeds R, Esteban P. Enhancing engineering project management through process alignment. *Eng Manag J*. 2021:1–19.
44. Malik M, Sarwar S, Orr S. Agile practices and performance: examining the role of psychological empowerment. *Int J Proj Manag*. 2020.
45. Bianchi M, Marzi G, Guerini M. Agile, Stage-gate and their combination: exploring how they relate to performance in software development. *J Bus Res*. 2020;110:538–553.
46. Rother M, Rosenthal M. An Approach to Becoming Agile in a Dynamic World. Helping employees develop scientific thinking empowers them to solve problems and make decisions. *AME: TARGET*. 2018:34.
47. Cohen I, Iluz M, Shtub A. A Simulation-based approach in support of project management training for systems engineers. *Syst Eng*. 2014;17:26–36.
48. Baron C, Daniel-Allegro B. About adopting a systemic approach to design connected embedded systems: a MOOC promoting systems thinking and systems engineering. *Syst Eng*. 2020;23:261–280.
49. Verschuren P, Doorewaard H, Mellion M. *Designing a Research Project; 2*. The Hague: Eleven International Publishing; 2010.
50. O'leary Z. *The Essential Guide to Doing Your Research Project*. Sage; 2017.
51. Yin RK. *Case Study Research and Applications: Design and Methods*. 6th ed. SAGE Publications Inc; 2017.
52. Lavrakas PJ. *Encyclopedia of Survey Research Methods; 1*. Sage Publications Inc; 2008.
53. Hove SE, Anda B. Experiences from conducting semi-structured interviews in empirical software engineering research. In: *11th IEEE International Software Metrics Symposium (METRICS'05)*. 2005:10–23.
54. Guba EG, Lincoln YS. *Fourth Generation Evaluation*. Newbury Park, CA: Sage Publications Inc.; 1989.
55. Lincoln YS, Guba EG. *Naturalistic Inquiry*. Newbury Park, CA: Sage Publications Inc.; 1985.
56. Wahyuni D. The research design maze: understanding paradigms, cases, methods and methodologies. *J Appl Manage Account Res*. 2012;10:69–80.
57. Bryman A. *Social Research Methods*. 4th ed. New York: Oxford University Press; 2012.
58. Yin RK. *Application of Case Study Research*. 3rd ed. Thousand Oaks, CA: Sage; 2012.
59. Niiniluoto I. The aim and structure of applied research. *Erkenntnis*. 1993;38:1–21.
60. Pugh S. Engineering design–unscrambling the research issues. *J Eng Des*. 1990;1:65–72.
61. Provost F, Kohavi R. Guest editors' introduction: on applied research in machine learning. *Mach Learn*. 1998;30:127–132.
62. Paluch S, Antons D, Brettel M, et al. Stage-gate and agile development in the digital age: promises, perils, and boundary conditions. *J Bus Res*. 2020;110:495–501.
63. Rhodes DH, Valerdi R, Roedler GJ. Systems engineering leading indicators for assessing program and technical effectiveness. *Syst Eng*. 2008;12:21–35.
64. Niederman F. Project management: openings for disruption from AI and advanced analytics. *Inf Technol People*. 2021. https://doi.org/10.1108/ITP-09-2020-0639

AUTHOR BIOGRAPHIES

Evelyn Honoré-Livermore is a Phd candidate at the Department of Electronic Systems at the Norwegian University of Science and Technology (NTNU). She received her MSc in Electronics Engineering in 2012 from NTNU, and a Master of Business Administration from Yonsei University in Seoul in 2017. Evelyn has experience as a project manager and systems engineer from the industrial space sector (2012–2017). She is researching systems engineering and project management methods for academic research projects. She is also the project manager of the small satellite HYPSO (www.hypso.space).

KNUT ROBERT FOSSUM works as a research manager at Centre for Interdisciplinary Research in Space - CIRiS, NTNU Samfunnsforskning AS. He holds a MSc in biotechnology and PhD in Production and Quality Engineering. Fossum has worked with European space sector since 1997, mainly related to integration and operation of ISS payloads. He has 15 years' experience with space project management and 7 years with research- and human resources management responsibilities. He was responsible manager for the definition and establishment of N-USOC (www.n-usoc.no) and CIRiS (www.ciris.no). From 2010–2017 he served as senior advisor and national delegate to the European Space Agency (ESA). Resent work and field of interest is related to the integral role of human dependability for the safe, reliable and efficient organization and management of complex sociotechnical projects, in specific the application of System Engineering methodology for development and operation of autonomous systems.

ERIK VEITCH is a PhD candidate at the Department of Design at the Norwegian University of Science and Technology (NTNU). He is responsible for applied research in the field of human-system integration for autonomous marine systems. His research focus is on understanding operators' situation awareness needs during remote intervention operations of autonomous surface vessels. Erik has a Master's in Ocean and Naval Architectural Engineering from Memorial University of Newfoundland (2018, Fellow of School of Graduate Studies) and industry experience as a marine hydrodynamics engineer at a private consulting company (2013–2016).

How to cite this article: Honoré-Livermore E, Fossum KR, Veitch E. Academics' perception of systems engineering and applied research projects. *Systems Engineering*. 2021;1–16. https://doi.org/10.1002/sys.21599

Paper J

Fossum, Knut Robert; Honoré-Livermore, Evelyn; Veitch, Erik; Palmer, Erika; Haskins, Cecilia "Towards an Integrated Project Complexity Narrative – A Case Study of Academic Organizations," Accepted to be published in Systems Engineering, (The Journal of The International Council on Systems Engineering), Wiley Periodicals, (2022) [21].

In this paper we look at how academics view complexity based on 18 semi-structured interviews.

Systems Engineering

Towards an integrated project complexity narrative – A case study of academic organizations

Journal:	*Systems Engineering*
Manuscript ID	SYS-21-046.R2
Wiley - Manuscript type:	Regular Article
Date Submitted by the Author:	n/a

Complete List of Authors:	Fossum, Knut; NTNU Samfunnsforskning, CIRiS Honoré-Livermore, Evelyn; Norwegian University of Science and Technology Faculty of Information Technology Mathematics and Electrical Engineering, Department of Electronic Systems Palmer, Erika; NTNU Samfunnsforskning AS, Veitch, Erik; NTNU, Department of Design Haskins, Cecilia; NTNU, Mechanical and Industrial Engineering

Keywords. One of these must be chosen at the time of submission::	SEE29 Other Systems Engineering Enablers, project management, Project Complexity
Application Area:	AS09 Other Application Sectors

Towards an integrated project complexity narrative – A case study of academic organizations

ABSTRACT

The last decade has seen a growing interest in the benefits of applying project management (PM) and system engineering (SE) in an integrated way towards complex projects and programs. The concept of project complexity dimensions, with roots in both disciplines, is suggested as a component of an integrated project complexity narrative. This paper investigates how such project complexity narrative is reflected when informants talk about the role of PM and SE in two academic organizations. Most informants address uncertainty and social-political risks as part of their work, but any consistent use of a project complexity narrative is related to environmental and technical systems. Findings also indicate difficulty differentiating between the concepts of complicated and complex. The paper further contemplates how these findings inform efforts to manage complex research projects and programs.

K E YWOR D S
Project complexity, project management, systems engineering, academia

1 | INTRODUCTION

Project-based research has become the prevailing practice for funding and organizing research efforts, and collaborative research projects have emerged as a particular form of academia–industry interaction[1]. This projectification of academia[2] adds universities to the list of organizations that conceive, design and undertake complex projects. Industries and society are facing increasing connectivity to systems, both technical and social, outside traditional controls. We need to have appropriate language, constructs, and organization to deal with emergent behaviors and usages that are expanded beyond original designs of system components.

Now, more than ever, successful research projects rely on the capacity to manage interactions between people, organizations, technology, stakeholder politics and business interests in a cohesive and holistic manner. Concurrently, these challenges associated with complex research projects have received limited research attention and theory development, creating a research gap.[3-5]

INCOSE's *Systems Engineering Vision 2035*[6] specifically calls out that managing complexity is a significant factor that requires new skillsets. Development of such new skillsets, and mindsets, are of importance for those with intentions to apply systems engineering (SE) and project management (PM) to complex projects in an integrated way.[7,8] Prominent contributions towards integrated PM and SE include the combined team efforts of representatives from the Project Management Institute, the International Council on Systems Engineering, and the Massachusetts Institute of Technology (PMI/INCOSE/MIT) focused on integration at the program level,[8,9] and the INCOSE chartered working group (in 2016) focused on integration at both program and project levels.[10]

The identification of characteristics of complex systems and potential methods to deal with complexity in system development has received considerable attention.[11-14] Watson, Anway, McKinney, Rosser and MacCarthy[15] describe and define distinguishing characteristics that can be used to differentiate between complex and non-complex systems and discuss how complexity can be managed in light of these characteristics. Potts, Johnson and Bullock[16] provide a review of literature related to challenges in system complexity evaluations and discuss challenges involved in operationally embedding complexity evaluations within an organization. For example, does the organization evaluate the technical system to be developed, the project to realize the technical system or both? How is the boundary of the system of interest (SoI) defined; is it limited to the technical system interfaces, the environmental context of the implemented system or does it also include an extended strategic and business context?

The seminal analysis within the project complexity research field by de Rezende, *et al.*[5] suggests that project complexity is defined by dimensions that include structural, uncertainty, novelty, dynamics, pace, social-political, and regulative. Sheard and Mostashari[17] provided one of the early explorations of relationships between such types of complexity, i.e., three types of structural complexity (size, connectivity, and architecture), two types of dynamic complexity (short-term and long-term), and socio-political complexity. Rebentisch and Prusak[8] (p.349) provide a Call to Action for academic organizations that emphasizes the role of the individual faculty members. Only by living and embodying the transformation can faculties demonstrate to students the criticality of being interprofessional and interdisciplinary.[18]

The objective of this paper is to investigate the extent to which academics draw upon a project complexity narrative when they talk about PM and SE in their work and discuss how these findings inform efforts towards the management of complex research projects and programs. In doing so, this paper does not compare or set integration of PM and SE in industry and academic settings against each other because the conceptualization of project complexity dimensions as part of a project complexity narrative is arguably applicable in both academic and industrial settings. In this paper, we present the results of a case study to achieve this objective while accounting for relevant contextual factors of academic organizations in general and academic organizations with Scandinavian socio-cultural contexts specifically.

The following section provides some contextual aspects of academic research organizations relevant for the project complexity discourse. This section also introduces the project complexity discourse in view of the larger system complexity literature, focusing on the concept of project complexity dimensions.

Section 3 describes our methods and research approach. Section 4 presents the analysis of the interviews. Section 5 discusses the findings and how they inform efforts to manage complex research projects and programs. Section 6 presents the conclusions and recommendations for future work.

2 | Background

Siegenfeld and Bar-Yam[12] propose that problems arise from mismatches between the complexities of a task to be performed and the complexities of the system performing that task. This multiscale version of the Law of Requisite Variety[19] is illustrated in Bar-Yam[20] by considering military conflicts. Siegenfeld and Bar-Yam[12] use the organization of academic departments as a further example of such mismatch, i.e., that subdivisions within the problem do not match the subdivisions within academia.

The proliferation of interdisciplinary research centers and collaborative research initiatives represents a response to this mismatch. Known management challenges for collaborative research include facilitation of mutual learning, enabling shared goal definition, creating rules for cooperation and synergy, managing heterogeneity, planning integration, and balancing personal attitudes and careers of the involved researchers.[21] Subdivisions in academic organizations made to address a specific type of complexity problem may influence other types of complexity, e.g., high uncertainty in goals or methods may result in more changes during the project. Increased change may increase the dynamic complexity, which again may bring increased structural complexity. High structural complexity of the organization may increase the socio-political complexity.[22]

Project management methodologies, such as network charts and Gantt charts, are arguably a form of systems thinking and govern how projects relate to complexity. However, Fowler, *et al.*[2] found a schism in the application of such PM methods in academia. While formal PM methodology and terminology were used by specially appointed research managers as a structure for reporting to funding agencies and other external parties, most researchers carry out their work without applying PM methods. Rather, they approached the work without much coordination or planning, in a so-called "fuzzy" manner.

Thus, while academic organizations are themselves complex, and many science disciplines address complex problems and have developed theories and methods to deal with complexity, there is limited evidence of applying PM or SE techniques to manage academic research projects, or programs, and their complexities.

To provide the context for the case study presented in this paper, in the following sections, we position the project complexity discourse in view of the larger system complexity discourse (section 2.1) and provide an account of project complexity dimensions and their roots in system and project complexity discourse (section 2.2).

2.1 | Project complexity and the system complexity discourse

Throughout its history, systems engineering (SE) has been the primary method for engineering in the face of complexity.[11] As such, systems engineering is the engineering of complexity.[23] The systems engineering discipline has evolved its practices via experientially developed principles and heuristics.[24] However, an expanding literature on networks, complexity, and complex adaptive systems theory has been developed to support the systems engineering approach to complex systems.[15,24-28] As with any expanding field of knowledge, the "theory – practice gap" is known, i.e., busy systems engineers and project managers rarely have the time to keep up with the literature, which often is diffused across the many interdisciplinary applications of complex systems science.[29]

An early description of complexity appeared in Weaver's seminal paper in 1948, titled: "Science and complexity", in which the concept of organized and disorganized complexity was introduced.[13,30] Since then, efforts have been made to identify different metrics of complexity in terms of description, creation and organization[31,32] and a review of "complex systems" definitions suggests terms such as statistical complexity, hierarchical complexity and disorder to characterize systems complexity.[33] Fischi, Nilchiani and Wade[34] review complexity in engineered systems and propose dynamic complexity measures to evaluate and compare system designs. Yang, Cormican and Yu[35] investigated how ontologies support SE and aimed to ascertain to what extent they have been applied. They further suggest a classification of SE knowledge areas where system fundamentals feature as a key SE knowledge area, with system, behavior, complexity, and emergence as sub-groups.

Sauser and Boardman[36] offer the *systemigram* as a method for modeling complexity to achieve a shared mental model[37] based on the soft systems methodology (SSM) of Checkland[38] The SSM and systemigram are often used to model systems in which the complexity has a sociotechnical aspect, or structural and dynamic complexity. The objective is to generate visualizations that reflect understanding of the needs of the stakeholders. Sauser and Boardman[36] review the different methods of visualizing system complexity. There are inherent limitations when modeling systems with pictures, links, and words, where the resulting model is a static representation and cannot easily characterize emergent behavior and uncertainties, especially when describing a System of Systems (SoS).

Potts, Sartor, Johnson and Bullock[39] discuss how reductionism is not sufficient for understanding complex systems because of the potential to lose the understanding of the system-as-a-whole, especially in the context of SoS. Some guiding principles for architecting complex systems using graph theory are suggested.[39] By addressing practical approaches towards complexity, complex systems and complexity science Simpson and Simpson[40], Sheard, *et al.*[11], Watson, *et al.*[15], Siegenfeld and Bar-Yam[12] and Grumbach and Thomas[41] make important contributions to bridge the "theory-practice gap" .

Rousseau[42] addresses the state of systems science, its relationship to complexity theory, and puts forward the need for a general systems theory to act as a foundation for systems engineering and systems practice. However, one can question if systems science needs SE more than SE needs systems science and general systems theory.[43] This implies the need to embrace a broader research agenda for systems engineering, including how we introduce SE to the future engineers.[35,44] By introducing a project complexity narrative that supports sense-making as a collaborative and iterative process one can sidesteps the challenges associated with the lack of any single, agreed definition of system complexity.[11,16]

The PM literature has several definitions of project complexity, but an agreed comprehensive definition is lacking, and no generally accepted framework has emerged to support the analysis of highly complex and innovative projects.[3,5] As such, complexity is an important and controversial topic in the PM discourse.[45] Brady and Davies[46] reviewed the PM literature on project complexity and further developed the framework introduced by Geraldi and colleagues[22] focusing on structural and dynamic complexity and how these might be managed in practice. Most of the PM discourse agrees that factors caused by unfamiliarity and the lack of knowledge ought not be associated with project complexity.[45] This aligns with a general acceptance, across important differences in epistemological orientation, of the notion of actor farsightedness[47], i.e., managers are expected to have a qualified view on how the future unfolds.

In their review of 420 scientific papers, Bakhshi*, et al.*[45] distinguish between three distinctive schools of project complexity: the Project Management Institute (PMI) view, the System of Systems (SoS) view and the Complexity Theories view.

The PMI view tends to focus on multiple stakeholders and ambiguity as two key characteristics of project complexity.[48] Until recently, the PMBOK[49] did not define or use the term "uncertainty", nor did it mention "complexity". Most researchers who tend towards the PMI view emphasize structural complexity, uncertainty and socio-political elements rather than other complexity dimensions.[45] The 2017 update of the PMBOK[50] introduces the PMI Talent Triangle® as part of its effort to ensure that its certifications are relevant to the needs of industry and organizations.[51] These expectations towards the skill sets of project managers reflected in the talent triangle arguably support efforts to decode project complexity. The PMBOK 7th Edition released in 2021 introduces a further shift from "process-based project management" to "principle-based project delivery", further aligning PMI to the changing dynamics of the management profession.

The shift from a process and predictability focus towards a dynamic and adaptability[52,53] focus is likely motivated by the increasing number of organizations confronted with challenges of engineering complex System-of-Systems (SoS), or engineering a system that operates in a complex SoS context.[6,54] There are several definitions of SoS that depend on the particularity of the application area.[5,55-57] System of Systems engineering (SoSE) has been considered by some as an opportunity for the SE community to define the complex systems of the twenty-first century.[57] In general, SoSE requires considerations beyond those usually associated with engineering to include socio-technical and sometimes socio-economic phenomena.[56] However, Ireland[58] suggests that any important contribution of research in SoSE has been based mainly in technology domains and relatively neglected the social and political areas.

A key aspect that distinguishes the SoS view from the PMI view is the lack of centralized control in managing autonomous and independent systems, both technical and organizational. Interest for the SoS view within the project complexity discourse have been rapidly increasing, and SoSE are employed in many large industries.[45,56] This view on SoS, as an approach to complex systems and projects, finds support in Cynefin,[59,60] a sense making model that proposes four categories: obvious, complicated, complex, and chaotic. As such, the Cynefin framework provides a potent tool to distinguish the complicated projects from the complex projects.

"The Complexity Theories view" is the term Bakhshi*, et al.*[45] used to group research papers that did not fit within the PMI and SoS view. This category accounts for a multitude of research that considers project or system complexities using various theories, e.g., contingency, network, chaos and complexity theory.[45] Most characteristics discussed in their research are time-dependent, observer-dependent, and problem-dependent, and as such difficult to further synthesize and generalize. It may seem that much of this research is motivated by a growing realization that classical PM techniques, e.g. breakdown structures, network analysis, Program Evaluation Review Technique (PERT) and critical path analysis, are most effectively applied in obvious or complicated problem contexts.[59,61]

In their bibliometric network analysis of 50 years of project complexity research de Rezende, *et al.*[5] conclude that PM research is changing from project control to project adaptability when dealing with complex projects. This is aligned with the call to complement mechanistic and modernist views with their false promises of prediction, certainty and control,[51] with a worldview that is made up of interconnected technical and social entities that more often produce behaviors that cannot be predicted by analyzing the behavior of a single part in isolation or by simply aggregating the behavior of the parts.[54,62] Addressing complexity organically rather than mechanistically represents such a shifting view.[55]

de Rezende and Blackwell[56] use seven dimensions to define and introduce a project complexity framework to allow researchers and practitioners to better understand projects and make more informed decisions. The following section elaborates on these different views on project complexity dimensions.

2.2 | Project complexity dimensions

The notion that a conceptualization of project complexity dimensions represents a contribution towards an integrated project complexity narrative for PM and SE disciplines springs from the roots that these concepts have in both disciplines, i.e., in context of both system and project complexity discourse. However, the concepts usually are not referred to as dimensions. Sheard and Mostashari[17] and Sheard and Mostashari[63] explored specific measures of complexity that could be compared and tracked to identify and mitigate risks in complex systems or development programs. They proposed a framework that includes structural complexity, dynamic complexity and socio-political complexity.[64] Others have also provided extensive reviews of different definitions further highlighting the diverse conceptual landscape,[31,33] The notion of referring to these concepts as dimensions can be attributed to Geraldi*, et al.*[22]. They described complexity of projects in five dimensions; *structural*, *uncertainty, dynamics*, *pace* and *socio-political* complexity. In their seminal review, de Rezende, *et al.*[5] suggest that project complexity is defined by seven dimensions that include structural, uncertainty, novelty, dynamics, pace, social-political, and regulative. In the following we address the five dimensions of Geraldi*, et al.*[22] and propose to subordinate the novelty and regulative dimensions of de Rezende*, et al.*[5] to the uncertainty and socio-political dimensions respectively.

The concept of *structural complexity* made its first appearance in PM literature in the 1990's [65-67] and has since been accepted as a feature of project complexity. Geraldi, *et al.*[22] found size (or number), variety and interdependence to be key attributes of structural complexity. This aligns with the three types of structural complexity described by Sheard and Mostashari[17] i.e., size, architecture and connectivity. It is also the type of complexity that has seen the most extensive development of complexity metrics[16,68,69] and the concept with the most mentions in both project and system complexity literature. Structural complexity should be understood as applicable to both engineered systems and the organizations put in place to deliver them. Brady and Davies[46] compare the complexity of two successful construction megaprojects – the Heathrow Terminal 5 and the London 2012 Olympic Park - by considering differences in the approach to managing structural and dynamic complexity. They conceptualize structural complexity as the "arrangement of components and subsystems into an overall system architecture" and dynamic complexity as the "changing relationships among components within a system and between the system and its environment over time" (p.24).

As such, it is useful to characterize *dynamic complexity* as "a change in any of the other dimensions of complexity"[22] (p. 980). The attributes for dynamic complexity are less developed and specific than those for structural complexity, but dynamic behavior is a prevalent aspect of complex projects and often linked to uncertainty of variables. Fischi*, et al.*[34] address dynamic complexity measures for use in complexity-based system design and Sterman[70] bring several concepts, tools and examples of system dynamics to solve complex problems, including complex projects. Sheard and Mostashari[63] describe two types of dynamic complexity that suggest a distinction between sudden rapid change in system behavior (short-term) and changes in number and types of things and their relationships (long term).

Uncertainty, and its relationship to risk, has been present in the management literature for almost 100 years and was proposed as a component of project complexity[67]. Uncertainty can be defined as the result of not having accurate or sufficient knowledge of a situation.[71] Uncertainty about project inputs affects modeling, evaluation and control of projects and establishing the objectives of time, cost, quality, and safety. Uncertainty is also found when there are unknown variables of the project output[65], e.g., in "mega-projects"[72] or in research projects.[73] The system complexity discourse includes different views on the question of whether complexity is, or should be understood as, observer dependent, or not.[74,75] It is not our intention to answer this question here, but uncertainty is a project complexity dimension that addresses this point when asking "uncertain for whom?" This is also why we assert that novelty is better addressed as an aspect of uncertainty, rather than to conceptualize it as a separate project complexity dimension, e.g., as de Rezende*, et al.*[5] suggest. When considering uncertainty as a dimension of project complexity we suggest rating the uncertainty on a scale from highly intrinsic to highly contextual.

Pace, together with structural complexity, represents a tangible construct with several commonly accepted indicators. It essentially refers to the rate of planned delivery of projects or systems, i.e. urgency and criticality.[76] Often it is difficult to operationalize metrics since pace is always relative to some reasonable or optimal measure. What is reasonable, or rational, is relative to goals and context.[77] Increasing pace by delivering systems sooner (e.g., 1 year instead of 2 years) can result in increasing complexity, while introducing and pacing iterations could help smooth aspects of the other complexity dimensions.

Sheard and Mostashari[17] consider *socio-political complexity* as the effect of individuals or groups of people on complexity and include sociological phenomena, such as fads and marketing, or the fields of economics, environmental sustainability, and politics. They suggest that the primary rationale to group these phenomena together is that most engineers have neither the education nor aptitude to deal with them. In PM discourse, the socio-political dimension of complexity was introduced by Geraldi and Adlbrecht[78] and Remington and Pollack[79], and it is considered a key area of skillsets that project managers need to develop to manage effectively.[80] The socio-political dimension of project complexity is frequently related to stakeholder engagement, both project internal and project external stakeholders.[81,82] Socio-political complexity more often relates to decisions regarding "doing the right project" rather than "doing the project right." The socio-political dimension also includes "behavioral complexity" emerging from the interactions between people within organizations, involving aspects such as transparency, empathy, variety of languages, cultures, disciplines, etc.[22] While de Rezende*, et al.*[5] suggest regulative, i.e. control or directive according to rule, principle, or law, as a seventh project complexity dimension, we suggest including such aspects as sub-groups of socio-political complexity dimension: Interpersonal/behavioral, societal/political and organizational (intra- and inter-). Although socio-political complexity is straightforward to broadly conceptualize, it is complicated to operationalize and is often considered as a cradle for "wicked problems." [81]

This notion of project complexity dimensions represent a compromise between a paralyzing holistic view and an over-simplified reductionistic view of complexity.[22] The notion of project complexity dimensions does not contradict the theories of complexity. Rather it enables more precise sense making and collaborative description, which will lead to a more informed approach to managing the complexities of projects and systems.

3 | RESEARCH APPROACH

This paper reports from an ongoing effort to investigate the perceptions and application of systems engineering (SE) and project management (PM) in academia.[83,84]

The research objective of this paper is to investigate to what extent academics draw upon a project complexity narrative when they talk about PM and SE in their work and discuss how these findings inform efforts towards management of complex research projects and programs.

The research objective and following discussion of the results are informed by the systematic literature review presented in section 3.1. The empirical results in this paper originate from case studies in two Scandinavian organizations (Section 3.2). Section 3.3 accounts for the reliability and validity of the research approach.

3.1 | Literature review

The notion of developing new valuable contributions towards system engineering was triggered by a literature review in this journal. Relevant research papers were identified using the keywords "system complexity" (18 papers) and "project complexity" (7 papers). The comprehensive literature reviews by Bakhshi, *et al.*[45] and de Rezende, *et al.*[5] were used to select relevant project complexity papers, i.e. by the "snowballing" method. The inspiration towards an integrated PM and SE approach has its roots in research and literature generated from a joint PMI, INCOSE and MIT project and documented by Rebentisch and Prusak[8] and the INCOSE chartered working group (in 2016) focused on integration at both program and project levels.[10]

3.2 | Interviews – data collection

The data were collected from semi-structured interviews of 45-50 minutes with 18 informants. The number of informants adequately represent the boundaries of the case study. To ensure this, we used a key-informant sampling method [85] that guided the selection of participants based on their involvement in space projects at two academic institutions. The space projects undertaken by the different groups mainly focus on technology development in applied research. Nine informants are employed at an independent research institute and the other nine are employed at the faculty of engineering at a public university. The informants all have a M.Sc. degree, and most have Ph.D. degrees in either natural science or engineering, but not SE or PM. The organizations are in the same region and influenced by Scandinavian socio-cultural norms. The informants were anonymized, and interviews were transcribed in their given language, either Norwegian or English.

See Table 1 in the appendix for the semi-structured interview guide. The informants were not asked questions that contained the word complicated, complex, or any grammatic variation of complexity, emergence, or dynamic behavior.

The interviews and transcripts are available in the informants' mother tongues but for the purpose of this paper the excerpts presented are translated to English.

3.3 | Reliability and validity

Lincoln and Guba[86] introduced four criteria for research trustworthiness commonly applied among social science researchers to attribute reliability and validity to the specific nature of qualitative research. In qualitative research, *dependability* is often used similarly to reliability in quantitative research while *credibility, transferability and confirmability* are considered in the research design and data collection as consistent with internal validity, external validity and construct validity in quantitative research.[87] The use of case study as a methodological approach has some inherent limitations towards transferability, i.e., limitations toward the extent to which the findings can be analytically generalized to other situations.[88]

4 | Case study results: Analyzing the interviews

The findings are presented in five sections that present the authors' consensus evaluation of the informants' answers regarding the position of SE and PM in their work.

The analysis indicates to what degree the five complexity dimensions, i.e., *structural, uncertainty, dynamics, pace and socio-technical*, are represented in the answers. All quotations reflect the opinions of the respective informant and do not necessarily reflect the views of the authors.

Each section includes a table with results from a search in the interview transcripts for relevant key words. Words are only included if they are used in context of projects, programs, systems, engineering or organization.

4.1 | Structural complexity

When informants were asked to reflect on SE, its application and the meaning of the term, the typical answers were related to aspects of structural complexity, specifically the benefits of organized and holistic approaches to complicated systems. For example, Subject 1 described how SE is concerned with seeing the whole system holistically and reducing it into manageable parts to identify and increase the understanding of relationships between its parts, while still maintaining the overall overview.

> Subj1*: So, it [Systems engineering] is about seeing the whole system, but in a more holistic way where you reduce the system into manageable parts where you can see the relationships between the different systems and how they influence and interact with each other. But without it becoming a mess where everything is dependent on everything.*

Other answers also reflected a systematic, reductionistic view on how SE could support the management of information in complicated project deliveries.

> Subj2*: (…) a way to systematically manage large amounts of information. A type of methodology that can help you with that. To sort and prioritize complicated systems. If you must build or deliver something.*

It is interesting to note that none of the informants addressed aspects of structural complexity when asked about research projects, nor was the word "structural complexity" used by the informants. However, the characteristics associated with structural complexity were often mentioned by informants when prodded about their understanding of SE.

Table 4.1 Keyword search in the interviews relevant to the structural complexity dimension

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Structure	2	4	2		11	3	2			1				4		1		
Number			1				1									2		
Size	2			2	3	2	2	2	7	3	2					5	3	1
Architecture		1	8						2						1			

4.2 | Uncertainty

Uncertainty and relevant characteristics were often used by the informants. The intrinsic uncertainty about the outcome of research projects, and the resulting uncertainty about data and resource availability and planning for the next iteration / cycle of the research process, were all frequent aspects of the informants' answers, e.g., the resulting challenges of planning the availability of relevant personnel, laboratory resources and procurement of materials and technical services.

> Subj12*: Given the resources we have I understand that there is a lot of frustration. You want something to work straight away if it stops working. If you have an idea, you want the answer immediately. All waiting leads to frustration. As a whole, I think we [the department] have enough resources for technical support. You could discuss what is the optimal organization of the technical resources. How do you distribute them? That is a continuous discussion. But as a whole we cover the most important areas. The challenge is that it is a large department and there are many needs that should be satisfied concurrently. We have to figure out how to ensure that.*

There appears to be a demarcation line between how informants approach projects in the research domain and engineering domain. Decisions under uncertainty in the engineering domain are usually mitigated by selecting approaches where solutions can be validated and verified against requirements, i.e., the process aims to prove that the selected solution is good enough. In the research domain, there is never a final answer or solution, even though the research project finishes. There will always be a new problem or question to address with new methods or materials to push the research boundary.

Subj13*: This is how I distinguish between research and development. Development continues even though we are unsure if the prototype is good enough. You have to make a decision, do as well as you can. This is the available information, the available components – do as good as you can. Research is more like..."hmm..is it possible to make this a little tiny bit better" – let's work 5 years on making it a little tiny bit better*.

Table 4.2 Keyword search in the interviews relevant to the uncertainty dimension

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Uncertainty			1	1			1											
Cost			6		5	1	2						4				2	3
Control		4		1		2		2	1	2				1			1	
Risk			2		10	5		1		2				1			1	

4.3 | Dynamics

None of the questions asked resulted in relevant reflections around the phenomena of dynamics or adaptability. However, change was frequently mentioned. It is interesting to note that although several of the informants work with complexity and emergence within the field of cybernetics, and most of the informants are seasoned project managers used to handle socio-political aspects, dynamics did not come up in either the context of PM or SE.

Table 4.3 Keyword search in the interviews relevant to the dynamic dimension

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Change	1	1		10	6	4	4			5	2	2			4		6	4
Dynamic			2										1	1				
Variability																2	3	
Adaptability					1				1	2	1					1	1	

4.4 | Pace

Pace was frequently brought up by the informants. However, pace was predominantly addressed as a temporal phenomenon. Although several informants have experience with agile methods and processes, informants did not associate increased pace of cycles or iterations, e.g., use of rapid prototyping as an approach to cope with complexity.

Subj18*: And what I have tried now is to do this quicker (build a system). So that we can uncover what is wrong quicker. That it shouldn't take that long before something breaks. But to show, to be able to answer why we do this. Is this design so much better than that design? Some of the engineers working on this system will do incredible things. But then you have to explain that yes, if we had infinite time, we could do amazing things. Now, we try to do what is good enough. We want to make this measurement campaign. Or we try to do something amazing and never make the measurement campaign. And this trade-off with keeping people sufficiently enthusiastic and giving them enough freedom, while at the same time ensuring that they will deliver at some point.... that is hard in academia. People are used to getting what they want. I've experienced that we don't have good tools or methods. And no support. I have had to do everything on my own. We don't have an organization for this [building systems].*

Informants' answers indicate relatively relaxed attitudes towards milestones and deliveries, e.g., when comparing themselves with industry settings.

> Subj4*: In industry you are more concerned with ensuring that this will be a product with an actual reliable lifespan, also concerning maintainability and all the other costs associated with maintaining a product. We don't take that into account in research, we just want it to work. For a company, you would sign your own death sentence if you develop something you can't maintain and support during its lifespan.*

It is worth noting that informants were selected due to their association with space projects. Space projects often include hard deadlines such as a fixed, pre-paid, launch campaign. An increase in pace would arguably result in increased complexity. Several informants were positive towards shorter, often more defined projects.

> Subj1*: I would say that the timespan of this project is much shorter than the previous project I was involved with. It is easier to work in a project with an actual timespan. I think that is one of the definitions of a project, that it has a defined timespan. While something that was set up as a program, will not be a project. The other project turned into a program. (…) I think the mixture of trying to be a project and a program made it difficult to work with.*

Table 4.4 Key word search in the interviews relevant to the pace dimension

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Time	5	5	3	10	8	2	6	8	8	8	13	1	5	7	35	4	11	6
Frequency																		
Rhythm																		
Tempo																		
Speed						1												

4.5| Socio-political complexity

Informants frequently addressed topics related to socio-political complexity and the challenges they pose in any continuous research effort. However, there were limited reflections on the influence of PM or SE on a project generally or use of SE and PM tools and methods specifically, e.g., stakeholder analysis or "onboarding, to address this category of complexity.

Socio-political complexity is arguably the dimension with most challenges towards achieving a commonly agreed, operationalized ontology.[17,22,56] For simplicity, we analyzed the interviews with respect to three groups: Interpersonal/behavioral characteristics, societal/political characteristics and organizational (intra- and inter-) characteristics.

4.5.1| Interpersonal/behavioral

When the informants were asked how they perceive culture in their organizations, answers frequently addressed collaboration in teams of colleagues or interpersonal aspects derived from cultural norms and backgrounds.

> Subj17*: We have a Northern European work culture if that is descriptive. We have partners all over the world. And one notices that there is a different culture, especially when it comes to deliverables. There is a different way of complying to milestones in Northern Europe than in Southern Europe. Also, when it comes to replying to emails.*

4.5.2| Societal/political

When asked about their roles and responsibilities, informants addressed the socio-political complexity for research projects that actively involve the greater society as a stakeholder. Approaches towards socio-political complexity seem different than those used in structural or uncertainty dimensions. The reductionist and relativistic reasoning that is clearly present when addressing characteristics of structural complexity are lacking in this context. Responses suggest that the form and nature of communications to this category of stakeholder must be tailored to special interests that may, or may not, be the primary motivators for the project team.

> Subj4*: Yes, a lot [interaction with representative users]. There are many projects involving for example design students working with interaction design to students working with the politicians. We have a long road ahead of us. Convincing the politicians is probably the biggest hurdle. I am trying to disengage myself a bit right now to make sure we deliver on the technical side.*

4.5.3| Organizational (intra- and inter-)

When informants were asked how they perceive management within their own organizations and collaborating organizations, answers often addressed characteristics such as power, engagement, and support elements of socio-political complexity.

> Subj11*: There are many administrative hurdles, it is not easy to get administrative support for everything. … There are many things we need to figure out, and we don't get good support far down in the project organization. We have to organize everything about getting support at the labs. And that is not necessarily a part of the research, depending on how you view it. But it is not a part of research that we have to run around and get offers for manufacturing parts or for some equipment you need in the lab. Even though your research depends on it. And it is not very transparent how much money a project has or how much you have used. Only some people know this. And maybe we don't need to know that, but it would be nice to have a ballpark overview. And I've been involved in many projects where you have to do everything yourself. You don't get much administrative or technical support. I think that is the biggest challenge. Technical, judicial and financially it could have been more structured.*

Table 4.5 Keyword search in the interviews relevant to the socio-political dimension

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
People		12	23	3	9	1	6	6	8	12		6	4	16	17	2	22	8
Socio				2							1							
Political			3														2	
Language					3	1							1	1				
Communication	2		2	1	1	4		4	1	2		1			1	1		
Person*		2	3	2	3			7	1	3	4	3	3	2	3	2	4	7

5 | Discussion

There is a long history of strong heuristic SE and PM tools and processes that provide actionable insights by describing and analyzing complexity.[23,80] Interest in complexity and the use of holistic approaches by practitioners to manage complexity in systems and projects is not new.[89-94] The remaining significant challenges towards addressing the current research gaps identified by Shenhar, *et al.*[3] and Potts, *et al.*[16] are the lack of any single, agreed definitions of system complexity[11] and project complexity.[5]

We subscribe to an understanding of complexity as discussed by Sheard and Mostashari[17,24], i.e. complexity can be viewed as the inability to predict the behavior of a system due to large numbers of constituent parts within the system and dense relationships among them.[95] In complex projects and systems, these constituent parts may be technical, economic, social and cultural in nature and contribute towards *structural, dynamic, and social-political* complexity.[16,22,54] The large body of literature addressing structural, dynamics and socio-political complexities motivated us to follow Geraldi*, et al.*[22] and include two additional categories to conceptualize project complexity dimensions, i.e., *pace* and *uncertainty*.

A growing volume of discourses recognize that traditional dictionary definitions of complicated, complex, or chaotic, do not provide sufficient support toward describing and addressing contemporary problems that exhibit these attributes. A complex problem area requires a different approach than a complicated one. That is, solving a complex problem as if it were merely complicated risks delivering unsatisfactory solutions with low effectiveness and poor or inadequate performance. In turn, organizations that recognize these distinctions realize a better understanding of the interplay of scientific and heuristic pathways driving the emergence and evolution of system principles and methods across science and engineering fields. [26] We assert there are benefits, both practical and philosophical, for organizations to embrace an integrated project complexity narrative that addresses the challenges associated with contested definitions of systems complexity[15,68] and project complexity.[5,54]

We propose that as a first step towards embedding system and project complexity thinking within a wider learning cycle, academic organizations should focus on project complexity dimensions, rather than complexity factors, characteristics, or (contested and context dependent) definitions. This focus should enable lessons to be identified, learned and shared across disciplinary domains and thematic contexts.

The concept of dimensions is commonly associated with a measurable attribute of a particular kind, such as length, breadth, depth, or height, e.g., "the final dimensions of the system were 235 x 543 cm." Such structural aspects of complexity are connected more readily to some metrics. However, for aspects such as socio-political aspects of complexity one may need to apply expert judgement to assign any value or measure, e.g., number and type of stakeholders relevant for a project.

In the next section we discuss how the five complexity dimensions are reflected when our informants talk about PM and SE. In the section 5.2, we discuss how conceptualization of the project complexity dimensions contribute towards an integrated project complexity narrative for PM and SE, i.e., how it contributes towards a "better reflection on project complexity". In section 5.3, we discuss the trustworthiness of the research presented.

5.1 | Project complexity dimensions in our case studies

We found *structural complexity* to be the most familiar dimension for our informants. It is arguably the dimension where reductionistic theory and methods are most efficiently and effectively applied. While there are recognized SE and PM capabilities that lend themselves to be applied to the structural complexity dimensions, there are known barriers to employing them, such as a lack of training or culture.

The intrinsic *uncertainty* of the research process and the resulting challenges in timely planning and acquisition of needed resources could be seen as an example of the mismatch in the multiscale complexity of academic organizations.[12] The temporal aspect of planning the needed resources often does not match that of the iterative learning cycle, i.e., the *pace*, of most research processes. As such, academic organizations would benefit from an "organizing rather than an organization" focus. Introducing a shift from "farsighted" governance towards "spontaneous" governing would be one way of engaging uncertainty via increased dynamics**.** Increasing the capabilities to manage *dynamic complexity* in an organization could be one approach to engage both structural complexity and uncertainty.

Our findings also indicate that the concept of multiscale complexity of academic organizations[12], e.g., ambiguously coupled behavior, relationships, and structures on many scales, offers a novel approach to understanding known *socio-political* barriers towards SE and PM practices in complex research projects and programs.[21,73]

We suggest that conceptualization and application of project complexity dimensions support the development and comparison of individual mental models with a shared narrative for articulating our understanding of the dynamic and interconnected nature of a complex research project. However, since individual rationality is bounded in unique ways, depending on personal, cognitive capabilities, social and cultural background and professional training and experience, there will always be a point where the question: "for whom is this system too complex to comprehend and thereby to manage?" needs to be addressed.

5.2 | Towards an integrated project complexity narrative for SE and PM

Technical systems are composed of elements that can be described at various scales, e.g., materials, components, unit assemblies, subsystems. Likewise, academic organizations can be described at various levels of hierarchy, e.g., faculties, departments, research groups, support staff.

Understanding that behavior, relationships, and structure are not reducible to only one level but rather exist on many levels and are ambiguously coupled across multiple entities. As such, the hard problems often arise from mismatches between the complexities of a task to be performed, i.e., design and deliver a complex system, and the complexities of the system performing that task, i.e., project-based organizations.

Thus, to understand and communicate project complexity one should not start by focusing too much or too little on complexity, at any scale, but rather focus on the consideration whether the complexity of the project and program organization is tailored to address the complexity of the problem to be addressed and systems to be developed. We propose that an integrated narrative for project complexity with the five dimensions as a foundation would support SE and PM practitioners to become key facilitators for such efforts. Facilitators should be placed deliberately in the key positions required for successfully managing both system and project complexity in the wider organizational context.

Although this paper represents a very limited selection of organizations and informants because it is a case study, our research suggests a general lack of practice in identifying and discussing the implications of project and program complexity. Moreover, the language used by informants when talking about SE and PM suggests a critical gap in the understanding of SE and PM as disciplines with powerful potential towards coping with the challenging characteristic of project and system complexity. We suggest that the conceptualization of project complexity dimensions represent a potent platform for SE and PM disciplines to foster an integrated narrative for complexity, both for academics and for other organizations and practitioners.

The conceptualization should enable lessons to be identified, learned, and shared across organizational and discipline domains and contexts. There are some advantages associated with a low number of concepts, when introducing new initiatives in any organization. As such, the five project complexity dimensions represent an advantageous entry level to discussing both project and system complexity.

5.3 | Trustworthiness of the research

In line with Lincoln and Guba[86], Bryman[96] and Wahyuni[87] we discuss the dependability, credibility, transferability and confirmability of the results as a measure of the trustworthiness of the research.

Dependability, i.e., reliability that promotes replicability or repeatability, is an inherent challenge of case-based research, and in this study applies in equal measure to the evaluators, i.e., authors, stance and experience. However, the research approach reported in this paper allows for other researchers to reproduce the interviews, with their own set of participants. The *credibility* of the results was considered in the key informant sampling method and selection of the semi-structured interview guide. By evaluating the informants' answers for references to the different project complexity dimensions[54] and complexity characteristics[15] one can, with credibility, say something about the position of complexity thinking in relation to PM and SE in our case study organizations. The *transferability* of the research, i.e., the applicability of results into other settings or situations, have limitations linked to the characteristics of the case organizations, i.e., academic organizations with Scandinavian socio-cultural norms. However, our findings align with the larger project and system complexity discourse, and as such indicate a problem area, and corresponding solution space, that is applicable across sectors and jurisdictions. *Confirmability* of the study is methodologically sound but limited by decisions made due to both practical and legal aspects. The interviews and transcripts are only available in the participant's mother tongue, i.e., Scandinavian languages. Privacy regulations and nature of the consent given by informants also limits sharing the research data.

6 | CONCLUSIONS AND FUTURE WORK

Academic organizations display a high degree of multiscale complexity, from the individual researchers, research projects, the department's academic discipline, to their role as educators of the future workforce for society. As such, managing project complexity requires multiple perspectives dependent on the context. Project complexity dimensions represent potent concepts to initiate assessments of such context.

Our findings indicate that any consistent differentiation between concepts of complicated and complex is lacking. Furthermore, when addressing characteristics of project complexity informants focused on physical and logical systems. Although most informants address aspects of uncertainty and socio-political aspects of their work, such narrative challenges could inhibit groups from greater effectiveness in managing social-political risks in their work.

PM and SE are disciplines that were developed as a response to practical engineering and management challenges, and many of these challenges are symptoms of complexity. As such, PM and SE have been, and are, about coping with the complexity of our systems, organizations, and society. However, our findings suggest that PM and SE practices are not pervasive within academic organizations. This means that complex research projects and programs should not be studied solely based on an á priori assumption that there is a discrete set of organizational artefacts and actors formally associated with PM and SE governance.

Discourse on explorative projects and the role of system thinking in the context of academia-industry collaboration is a promising agenda and venue for further evolving multidisciplinary discussions on complexity, both in SE and PM. Future work should address the relationship between complexity dimensions, complexity characteristics and complexity factors, as well as ontology development. More exhaustive literature review, and larger, international surveys are required as part of the future work.

ACKNOWLEDGEMENTS

This work was supported by the Research Council of Norway, Statoil, DNV GL and Sintef through the Centers of Excellence funding scheme, Grant 223254 - Center for Autonomous Marine Operations and Systems (AMOS) and the Research Council of Norway through the IKTPLUSS program grant 270959 (MASSIVE). It has been approved by the Norwegian Centre for Research Data (project no. 560218).

DATA AVAILABILITY STATEMENT

The data are not publicly available due to privacy or ethical restrictions.

ORCID

Evelyn Honoré-Livermore https://orcid.org/0000-0002-5664-330X
Knut Robert Fossum https://orcid.org/0000-0002-1020-730X
Erik Veitch https://orcid.org/0000-0001-6049-8136

Cecilia Haskins https://orcid.org/0000-0002-2506-8808

REFERENCES

1. Brocke JV, Lippe S. Managing collaborative research projects: A synthesis of project management literature and directives for future research. *International Journal of Project Management.* 2015;33(5):1022-1039.
2. Fowler N, Lindahl M, Sköld D. The projectification of university research: A study of resistance and accommodation of project management tools & techniques. *International Journal of Managing Projects in Business.* 2015;8(1):9-32.
3. Shenhar AJ, Holzmann V, Melamed B, Zhao Y. The Challenge of Innovation in Highly Complex Projects: What Can We Learn from Boeing's Dreamliner Experience? *Project Management Journal.* 2016;47(2):62-78.
4. Lenfle S. Floating in Space? On the Strangeness of Exploratory Projects. *Project Management Journal.* 2016;47(2):47-61.
5. de Rezende LB, Blackwell P, Pessanha Goncalves, Denys M. Research focuses, trends, and major findings on project complexity: A bibliometric network analysis of 50 years of project complexity research. *Project management journal.* 2018;49(1):42-56.
6. INCOSE. *A world in motion – systems engineering vision 2025.* San Diego2014.
7. Hahn HA, Hodges AL. Integrating Program/Project Management and Systems Engineering in Practice. Paper presented at: INCOSE International Symposium2019.
8. Rebentisch E, Prusak L. *Integrating program management and systems engineering: Methods, tools, and organizational systems for improving performance.* John Wiley & Sons; 2017.
9. Conforto E, Rossi M, Rebentisch E, Oehmen J, Pacenza M. *Survey report: Improving integration of program management and systems engineering.* MIT Consortium for Engineering Program Excellence;2013.
10. INCOSE. PM-SE Integration Working Group. 2021; https://www.incose.org/incose-member-resources/working-groups/process/pm-se-integration.
11. Sheard S, Cook S, Honour E, et al. A complexity primer for systems engineers. *INCOSE Complex Systems Working Group White Paper.* 2015;1:1-10.
12. Siegenfeld AF, Bar-Yam Y. An Introduction to Complex Systems Science and its Applications. *arXiv preprint arXiv:191205088.* 2019.
13. Adcock R, Sillitto H, Sheard S. Compexity in SEBoK *The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 2.3* 2020; https://www.sebokwiki.org/w/index.php?title=Complexity&oldid=57988. Accessed January 15 2021.
14. Grogan PT. Perception of complexity in engineering design. *Systems Engineering.* 2021.
15. Watson M, Anway R, McKinney D, Rosser LA, MacCarthy J. Appreciative methods applied to the assessment of complex systems. Paper presented at: INCOSE International Symposium2019.
16. Potts MW, Johnson A, Bullock S. Evaluating the complexity of engineered systems: A framework informed by a user case study. *Systems Engineering.* 2020;23(6):707-723.
17. Sheard SA, Mostashari A. 7.3. 1 A complexity typology for systems engineering. Paper presented at: INCOSE International Symposium2010.
18. Goldsmith AH, Hamilton D, Hornsby K, Wells D. Interdisciplinary Approaches to Teaching. 2018; https://serc.carleton.edu/sp/library/interdisciplinary/index.html.
19. Ashby WR. Requisite variety and its implications for the control of complex systems. In: *Facets of systems science.* Springer; 1991:405-417.
20. Bar-Yam Y. Complexity of military conflict: Multiscale complex systems analysis of littoral warfare. *Report to Chief of Naval Operations Strategic Studies Group.* 2003.

21. König B, Diehl K, Tscherning K, Helming K. A framework for structuring interdisciplinary research management. *Research Policy.* 2013;42(1):261-272.
22. Geraldi J, Maylor H, Williams T. Now, let's make it really complex (complicated) A systematic review of the complexities of projects. *International Journal of Operations & Production Management.* 2011;31(9):966-990.
23. Honour EC. A historical perspective on systems Engineering. *Systems Engineering.* 2018;21(3):148-151.
24. Sheard SA, Mostashari A. Principles of complex systems for systems engineering. *Systems Engineering.* 2009;12(4):295-311.
25. Adams KM, Hester PT, Bradley JM, Meyers TJ, Keating CB. Systems theory as the foundation for understanding systems. *Systems Engineering.* 2014;17(1):112-123.
26. Rousseau D. A framework for understanding systems principles and methods. *Insight.* 2018;21(3):9-18.
27. e Silva SR. System Theoretic Process Analysis: A Literature Survey on the Approaches Used for Improving the Safety in Complex Systems. In: *Information Systems for Industry 4.0.* Springer; 2019:97-114.
28. Watson MD. Future of Systems Engineering. *INSIGHT.* 2019;22(1):8-12.
29. Bar-Yam Y. When systems engineering fails-toward complex systems engineering. Paper presented at: SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)2003.
30. Weaver W. Science and complexity. In: *Facets of systems science.* Springer; 1991:449-456.
31. Lloyd S. Measures of complexity: a nonexhaustive list. *IEEE Control Systems Magazine.* 2001;21(4):7-8.
32. Lopez V, Thomas D. Complexity assessment using SysML models. *Procedia Computer Science.* 2019;153:225-232.
33. Ladyman J, Lambert J, Wiesner K. What is a complex system? *European Journal for Philosophy of Science.* 2013;3(1):33-67.
34. Fischi J, Nilchiani R, Wade J. Dynamic complexity measures for use in complexity-based system design. *IEEE Systems Journal.* 2015;11(4):2018-2027.
35. Yang L, Cormican K, Yu M. Ontology-based systems engineering: A state-of-the-art review. *Computers in Industry.* 2019;111:148-171.
36. Sauser B, Boardman J. Systemigram Modeling for Contextualizing Complexity in System of Systems. In. Vol 276: Wiley Online Library; 2015:273-302.
37. Senge P. The fifth discipline: The art and practice of the learning organization. New York: Currency Doubleday. *SengeThe Fifth Discipline: The Art and Practice of the Learning Organisation1990.* 1990.
38. Checkland P. Researching real-life: reflections on 30 years of action research. *Systems Research and Behavioral Science.* 2010;27(2):129-133.
39. Potts M, Sartor P, Johnson A, Bullock S. Deriving Key Features of a System-of-Systems Complexity Evaluation Framework from an Industrial Case Study Analysis. Paper presented at: 2019 International Symposium on Systems Engineering (ISSE)2019.
40. Simpson J, Simpson M. Complexity reduction: a pragmatic approach. *Systems Engineering.* 2011;14(2):180-192.
41. Grumbach JL, Thomas LD. Integration principles for complex systems. *Systems Engineering.* 2020;23(6):684-706.
42. Rousseau D. A vision for advancing systems science as a foundation for the systems engineering and systems practice of the future. *Systems Research and Behavioral Science.* 2019;36(5):621-634.
43. Yearworth M. The theoretical foundation(s) for Systems Engineering? *Systems Research and Behavioral Science.* 2020;37(1):184-187.

44. Haskins C, Dahl HJ. Getting students hooked on systems engineering! Paper presented at: INCOSE International Symposium2013.
45. Bakhshi J, Ireland V, Gorod A. Clarifying the project complexity construct: Past, present and future. *International Journal of Project Management.* 2016;34(7):1199-1213.
46. Brady T, Davies A. Managing structural and dynamic complexity: A tale of two projects. *Project Management Journal.* 2014;45(4):21-38.
47. Sanderson J. Risk, uncertainty and governance in megaprojects: A critical discussion of alternative explanations. *International Journal of Project Management.* 2012;30(4):432-443.
48. PMI P. PMI's Pulse of the Profession In-Depth Report: Navigating Complexity. *Newtown Square, PA.* 2013.
49. PMBOK. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Fourth edition.* Project Management Institute, Incorporated; 2013.
50. PMBOK. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Sixth edition.* Pennsylvania, USA: Project Management Institute, Incorporated; 2017.
51. Boulton JG, Allen PM, Bowman C. *Embracing complexity: Strategic perspectives for an age of turbulence.* OUP Oxford; 2015.
52. Zhu H. Designing Systems with Adaptability in Mind. In: *Complex Systems Design & Management.* Springer; 2016:273-280.
53. Zhu H, Murray B, de Weck O, et al. Adaptability Metric Analysis for Multi-Mission Design of Manufactured Products and Systems. Paper presented at: INCOSE International Symposium2016.
54. Elia G, Margherita A, Secundo G. Project management canvas: a systems thinking framework to address project complexity. *International Journal of Managing Projects in Business.* 2020.
55. Serrador P, Pinto JK. Does Agile work?—A quantitative analysis of agile project success. *International Journal of Project Management.* 2015;33(5):1040-1051.
56. de Rezende LB, Blackwell P. Revisiting project complexity: a new dimension and framework. *The Journal of Modern Project Management.* 2019;6(3).
57. Henshaw M, Dahmann J, Lawson B. Systems of Systems (SoS). 2021; SEBoK v. 2.4, released 19 May 2021:https://www.sebokwiki.org/wiki/Systems_of_Systems_(SoS).
58. Ireland V. Research directions in SoSE. Paper presented at: 2015 10th System of Systems Engineering Conference (SoSE)2015.
59. Snowden D. "Cynefin: a sense of time and space the social ecology of knowledge management". In: Despres C, Chauvel D, eds. *Knowledge Horizons: The Present and the Promise of Knowledge Management.* Oxford: Butterworth-Heinemann; 2000:237-265.
60. Snowden DJ, Boone ME. A leader's framework for decision making. *Harvard business review.* 2007;85(11):68.
61. Snowden D. Complex acts of knowing: paradox and descriptive self-awareness. *Journal of knowledge management.* 2002;6(2):100-111.
62. APM. *Systems Thinking, How Is it Used in Project Management?* Princes Risborough: Association for Project Management;2018.
63. Sheard SA, Mostashari A. 6.2. 1 Complexity Types: From Science to Systems Engineering. Paper presented at: INCOSE International Symposium2011.
64. Adcock R, Sillitto H, Sheard S. Complexity. 2021; SEBoK v. 2.4, released 19 May 2021:https://www.sebokwiki.org/wiki/Complexity.
65. Turner JR, Cochrane RA. Goals-and-methods matrix: coping with projects with ill defined goals and/or methods of achieving them. *International Journal of Project Management.* 1993;11(2):93-102.
66. Baccarini D. The concept of project complexity—a review. *International journal of project management.* 1996;14(4):201-204.
67. Williams TM. The need for new paradigms for complex projects. *International journal of project management.* 1999;17(5):269-273.

68. Potts MW, Sartor PA, Johnson A, Bullock S. Assaying the importance of system complexity for the systems engineering community. *Systems Engineering.* 2020;23(5):579-596.
69. Sinha K, de Weck OL. Empirical validation of structural complexity metric and complexity management for engineering systems. *Systems Engineering.* 2016;19(3):193-206.
70. Sterman J. Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw-Hill Inc. 2000.
71. SEBoK_v.2.5. Uncertainty (glossary). 2021; https://www.sebokwiki.org/wiki/Uncertainty_(glossary). Accessed 14 October 2021.
72. Davies A, Dodgson M, Gann D. Dynamic Capabilities in Complex Projects: The Case of London Heathrow Terminal 5. *Project Management Journal.* 2016;47(2):26-46.
73. Lippe S, Brocke Jv. Situational Project Management for Collaborative Research Projects. *Project Management Journal.* 2016;47(1):76-96.
74. Luzeaux D, Ruault J-R, Wippler J-L. *Large-scale Complex System and Systems of Systems.* John Wiley & Sons; 2013.
75. Walden DD, Roedler GJ, Forsberg K, Hamelin RD, Shortell TM. *Systems engineering handbook: A guide for system life cycle processes and activities.* John Wiley & Sons; 2015.
76. Shenhar AJ, Dvir D. *Reinventing project management: the diamond approach to successful growth and innovation.* Harvard Business Review Press; 2007.
77. Mercier H, Sperber D. *The enigma of reason.* Cambridge Massachusetts, USA: Harvard University Press; 2017.
78. Geraldi JG, Adlbrecht G. On faith, fact, and interaction in projects. *Project Management Journal.* 2007;38(1):32-43.
79. Remington K, Pollack J. Tools for Complex Projects. Gower Publishing company. *aldershott, UK.* 2007.
80. Morris PW. A Brief History of Project Management. In: *The Oxford Handbook of Project Management.*2011.
81. Dentoni D, Bitzer V, Schouten G. Harnessing wicked problems in multi-stakeholder partnerships. *Journal of Business Ethics.* 2018;150(2):333-356.
82. Lehtinen J, Aaltonen K. Organizing external stakeholder engagement in inter-organizational projects: Opening the black box. *International Journal of Project Management.* 2020;38(2):85-98.
83. Fossum KR. *Global Collaborative Research Projects–Exploring scenario development*. Trondheim, Norway: Department of Mechanical and Industrial Engineering, NTNU; 2020.
84. Honoré-Livermore E, Fossum KR, Veitch E. Academics' Perception of Systems Engineering and Applied Research Projects. *Systems Engineering (Submitted).* 2021.
85. Lavrakas PJ. *Encyclopedia of survey research methods.* Sage publications; 2008.
86. Lincoln YS, Guba EG. *Naturalistic inquiry.* Vol 75. Newbury Park, CA: Sage Publications Inc.; 1985.
87. Wahyuni D. The Research Design Maze: Understanding Paradigms, Cases, Methods and Methodologies. *Journal of Applied Management Accounting Research.* 2012;10(1):69-80.
88. Yin RK. Validity and generalization in future case study evaluations. *Evaluation.* 2013;19(3):321-332.
89. Ramo S. *Cure for Chaos; Fresh Solutions To Social Problems Through The Systems Approach.* New York: David McKay Company, Inc; 1969.
90. Ramo S. *Century of mismatch.* New York: David McKay Company, Inc.; 1970.
91. Back KW. Decisions under uncertainty rational, irrational, and non-rational. *American Behavioral Scientist.* 1961;4(6):14-19.
92. Simon H. The Architecture of Complexity: Hierarchic Systems. *The Science of Artificial.* 1969.
93. Klein B, Meckling W. Application of operations research to development decisions. *Operations Research.* 1958;6(3):352-363.
94. Weaver W. Science and complexity. *American scientist.* 1948;36(4):536.

95. Sheard SA. *Assessing the impact of complexity attributes on system development project outcomes.* Stevens Institute of Technology; 2012.

96. Bryman A. *Social research methods.* New York, USA: Oxford university press; 2012.

APPENDIX A. Jane's story

The HYPSO project team consists mainly of temporary employees and students at NTNU, with little or no experience with aerospace engineering or multidisciplinary teamwork. The HYPSO project is a part of a larger strategic initiative from the faculty, to be able to deliver fast integration of scientific payloads into small satellites and autonomous assets. The project takes place in a university, and the students mainly do work in the projects through master and bachelor theses. The master students typically arrive in August/September and stay on until May/June next year. The bachelor students join the team in January and stay on until May. The students also have coursework in parallel with their theses, and cannot work full-time in the project. The professors supporting the project have not developed systems for space previously, and do not have the time nor knowledge to participate actively in the project. They do, however, provide supervising for the theses and give direction for the mission. The professors are also the main decision-makers since they are in charge of the funding of the project.

The team feels that there aren't enough people and resources in the project, especially in hardware (both electronics and mechanics). The people working in hardware feel stressed when thinking about the amount of work that has to be done, and how difficult it is to get the full picture. Even though the previous years' members are available by email and sometimes phone, they feel like they are not able to understand the full picture of the challenges or the work that needs to be done. The project manager/systems engineer (a PhD fellow) sees this and keeps getting questions that they feel like they told the team about several months ago, on where to find the information and how the decision was made. The PM understands that the team members are stressed, and thinks this is one of the reasons that these messages are forgotten/ignored. The PM feels like most of the team members are almost up to speed after six months, which is too long when they only have 9 months to contribute to the project in total.

Since the start of the project, there have been challenges in ensuring an understanding of decisions made and sharing these decisions. This first surfaced when some decisions were made, and the person in charge of them was away for a while, and it wasn't easy for the remaining team members to see why the decision had been made. The team re-iterated the decision with a more standard approach and communicated it in all meetings + documentation. The decision-making process has improved, but there is still a problem of making sure that everyone knows about the decision. The SE has tried setting up a “common model”, but it is not shared or used by all. The SE is also the PM, and a researcher, and feels like there isn’t enough time to be a good SE. The professors/stakeholders have not been able to do much about this since they haven’t been able to find someone to fill the role of either PM or SE to replace some of the duties.

Most decisions that have to be made are design-decisions and task planning decisions. The design-decisions are implemented in the common model after the decisions have been made, but the common model has not been used to support decision-analysis because there haven’t been enough resources to develop that capability. Most of the critical (=affecting many subsystems and mission) decisions are made with a minimum number of people, recorded in a minutes-of-meeting, and then documented in the design reports. However, there have been some instances where the team wasn’t able to see the breadth and impact of the decision made, where it has impacted the mission even though it was thought not to do.

The software part of the project is quite big and involves both pure algorithm development for future enhancement, firmware development for hardware, software interfacing between segments and subsystems, user interface development, and command & control software of the satellite and payload. It is both a large area and a complex area in the project. An additional complication is that many of the subsystems the payload interfaces with are developed by a different company, with limited documentation and parallel development/changing. There is one person in charge of the software group, who is also working as a researcher.

In March 2020, the team was going to make a go/no-go launch decision based on a Critical Design Review (CDR). The team decided on a set of requirements that would have to be verified by CDR, and if they weren’t, the decision should be to postpone the launch. Having this deadline helped focus on the development and integration testing, and documentation of these. The process up to CDR uncovered missing requirements and functions, and in some cases, lack of communication between the operations group and software group.

The CDR collocation meeting via Zoom was a great success. There were good discussions and the team got valuable feedback. The team was not able to verify all the requirements, and some critical dependability and risk issues were pointed out because of the lack of design maturity. After CDR, some of the momentum was lost. This was partly because of the COVID-19 pandemic, partly Easter holiday, and partly because students were refocusing on theses. This was evident both through lower participation in daily stand-up meetings, and from their feedback saying that they were focusing on writing thesis.

APPENDIX B. Models from MBSE

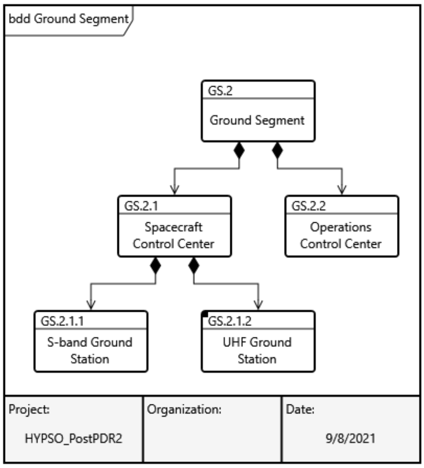


Figure B.1.: Block diagram of ground segment.

The whole model is not included because there are too many elements. However, some examples are included for information.

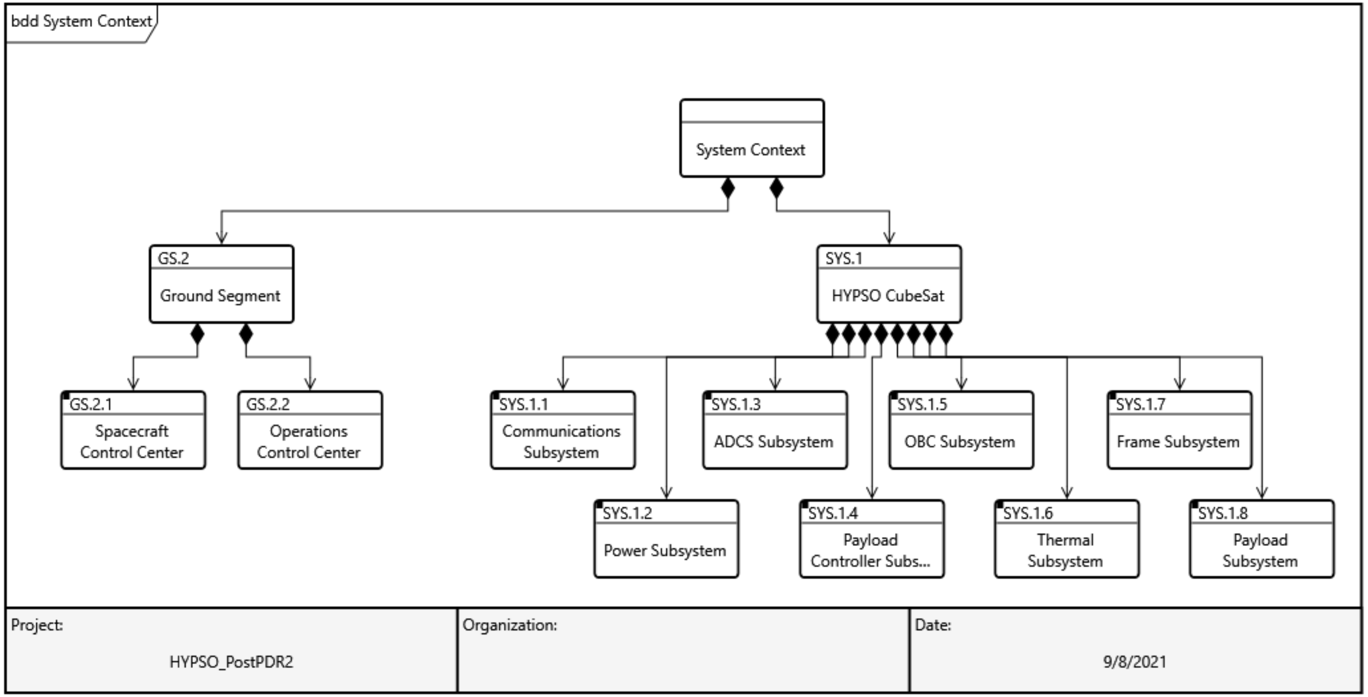


Figure B.2.: Block diagram of System Context.

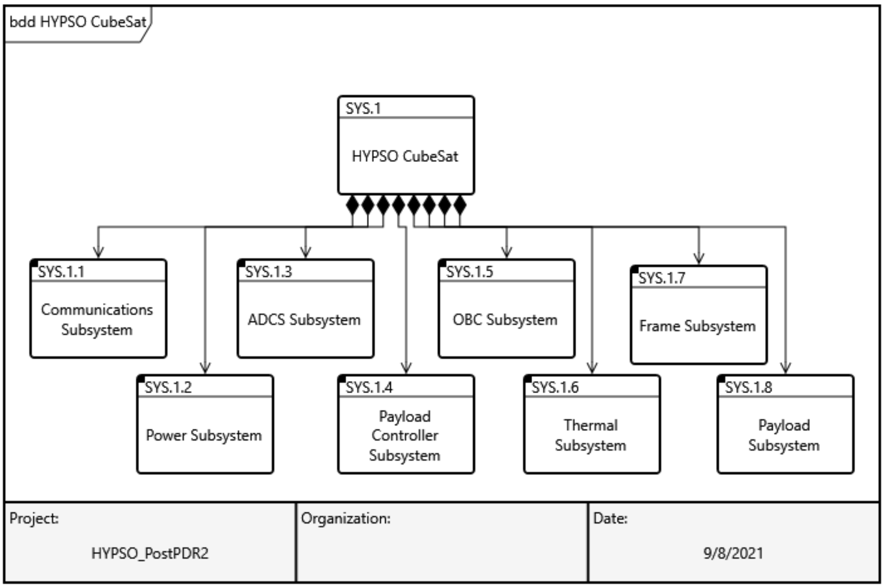


Figure B.3.: Block diagram of HYPSO CubeSat

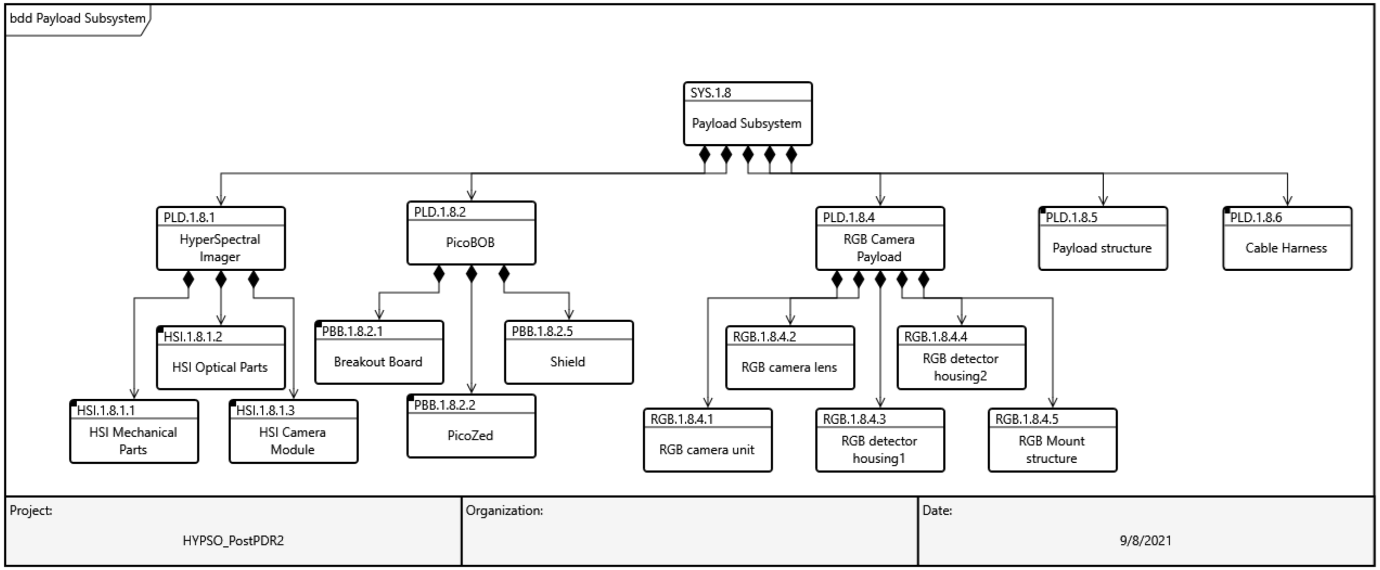


Figure B.4.: Block diagram of payload subsystem.

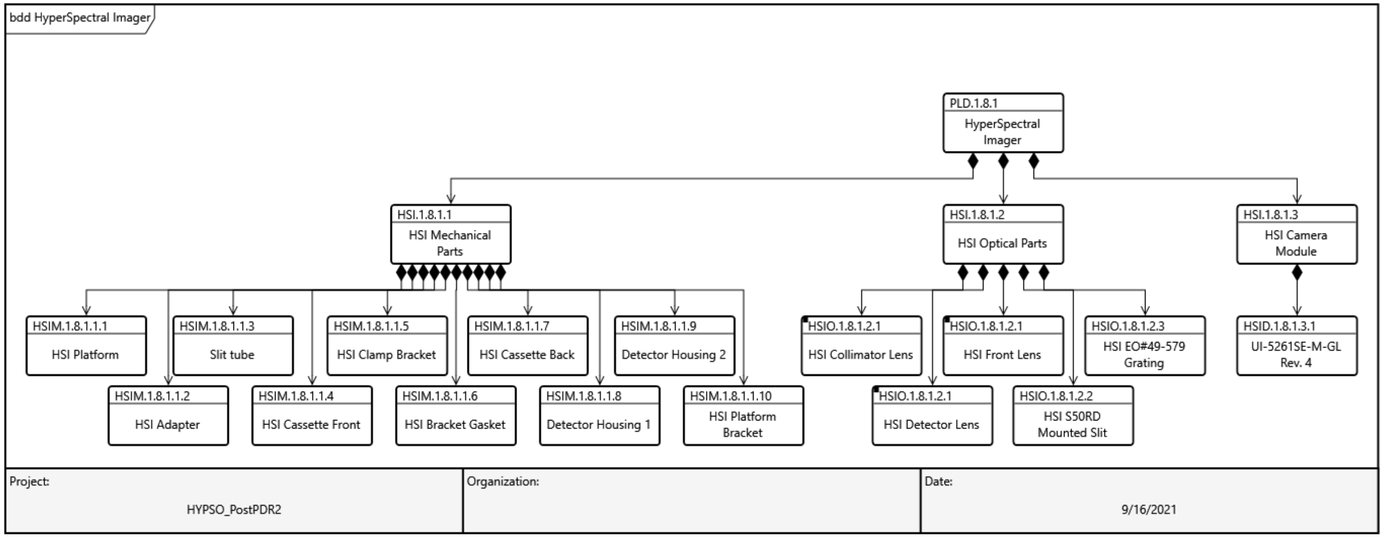


Figure B.5.: Block diagram of Hyperspectral Imager.

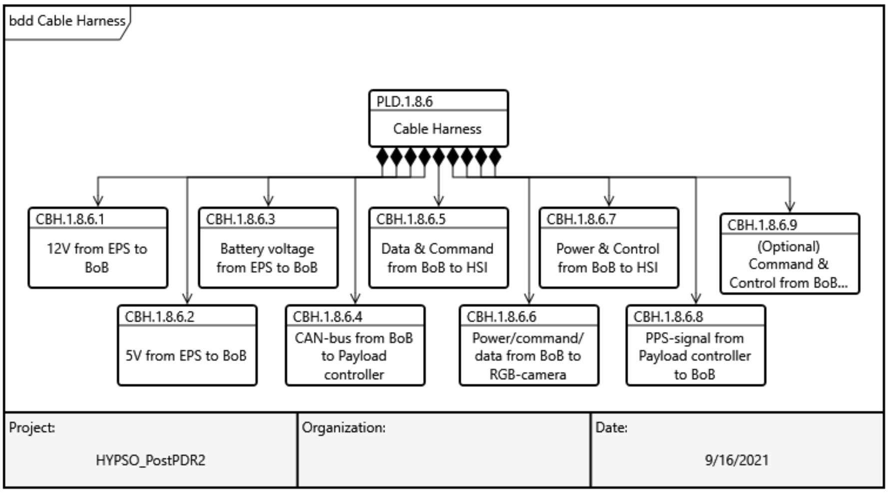


Figure B.6.: Block diagram of payload harness.

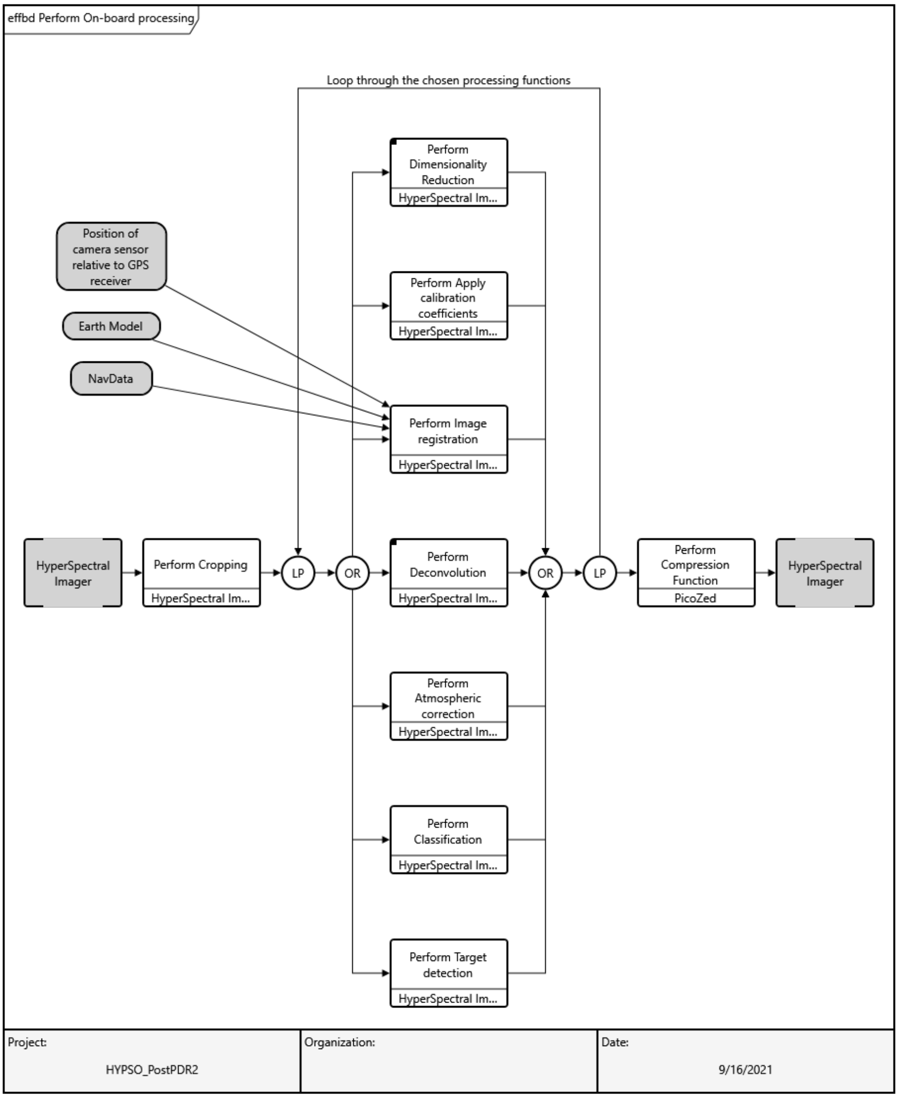


Figure B.7.: On-board processing as functional flow block diagram.

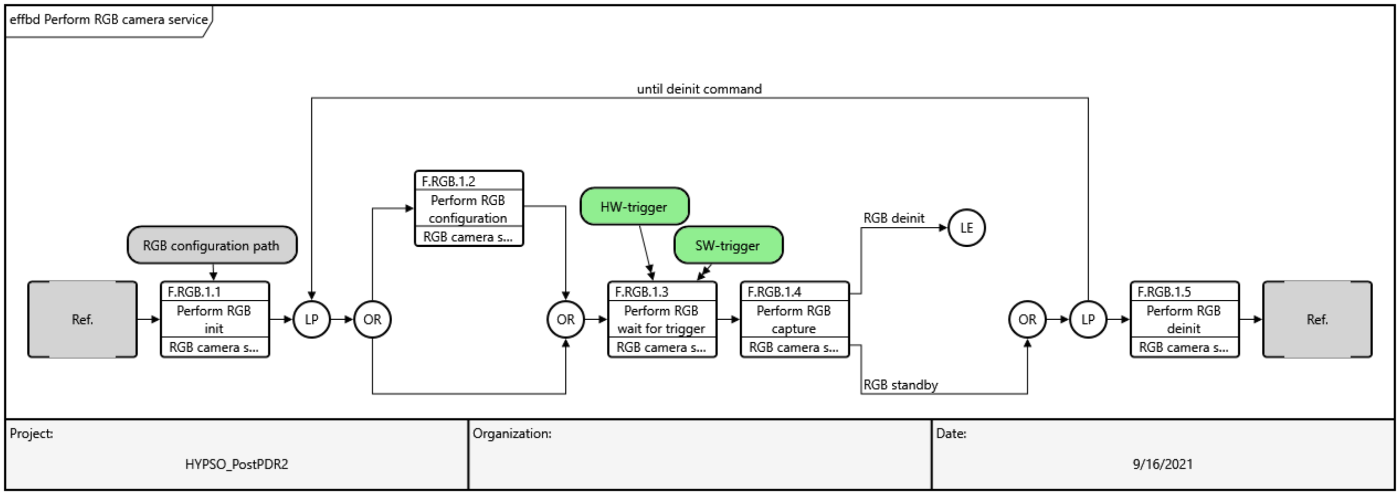


Figure B.8.: RGB camera service as functional flow block diagram.

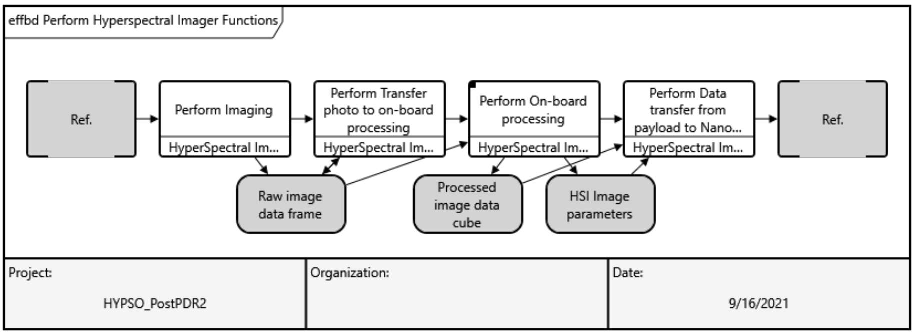


Figure B.9.: Hyperspectral Imager as functional flow block diagram. High-level only.

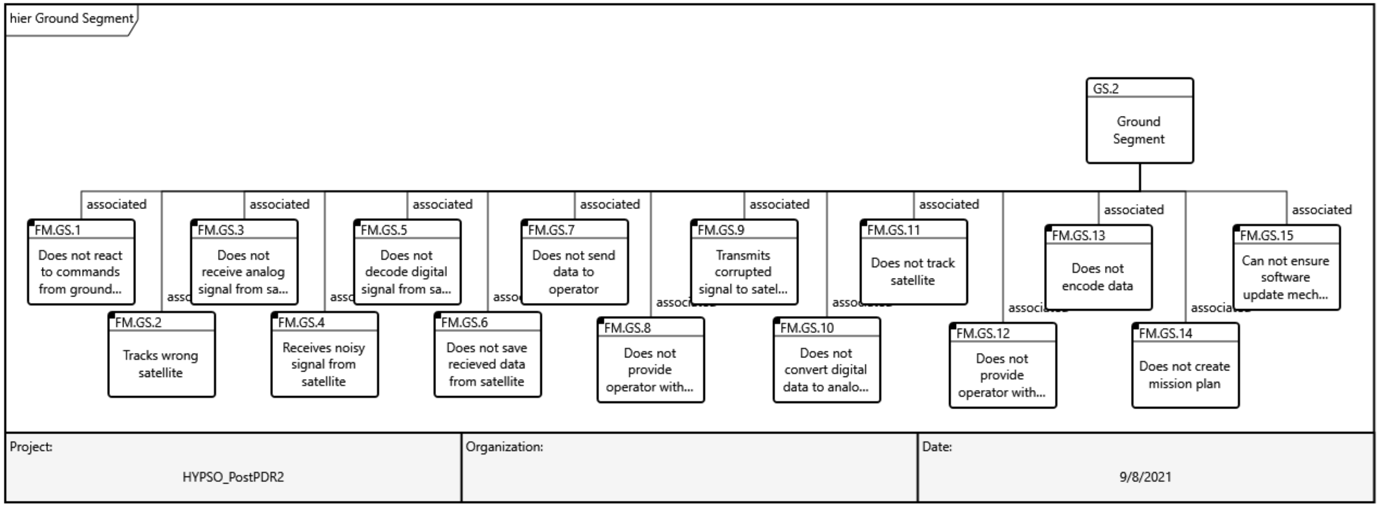


Figure B.10.: Ground segment failure modes.

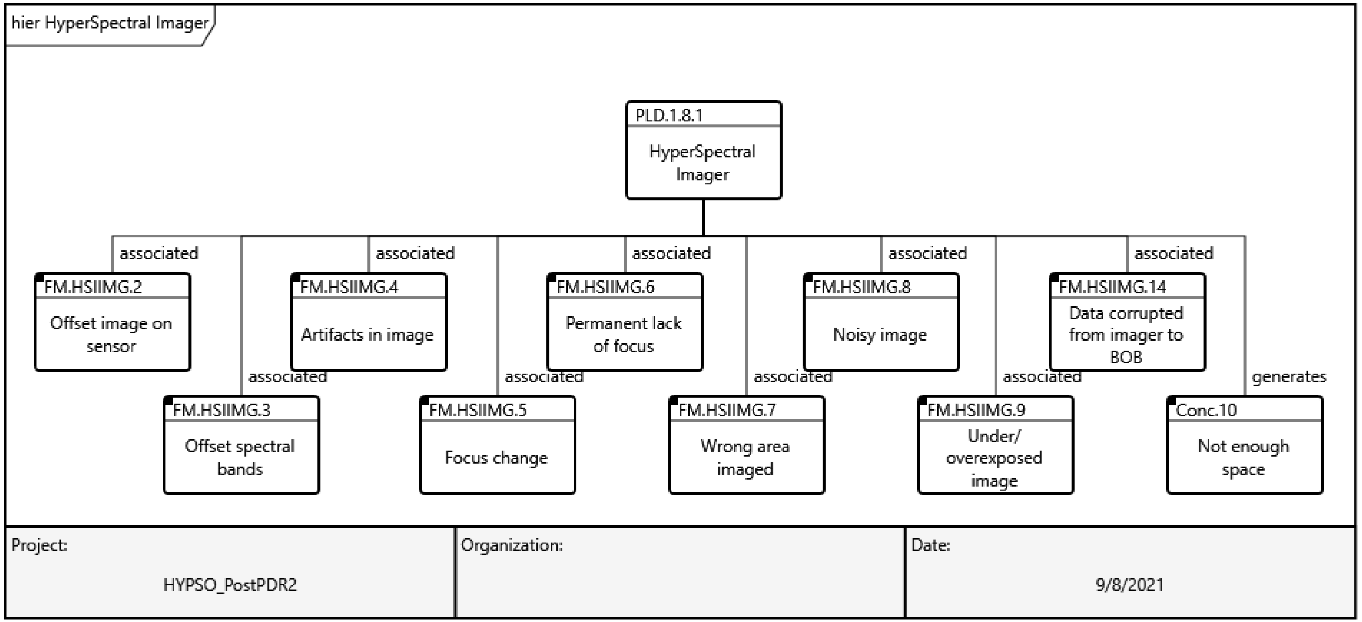


Figure B.11.: Hyperspectral Imager failure modes. Note: Conc.10 is a concern that is also associated with the HSI imager, regarding if it fits in the spacecraft.

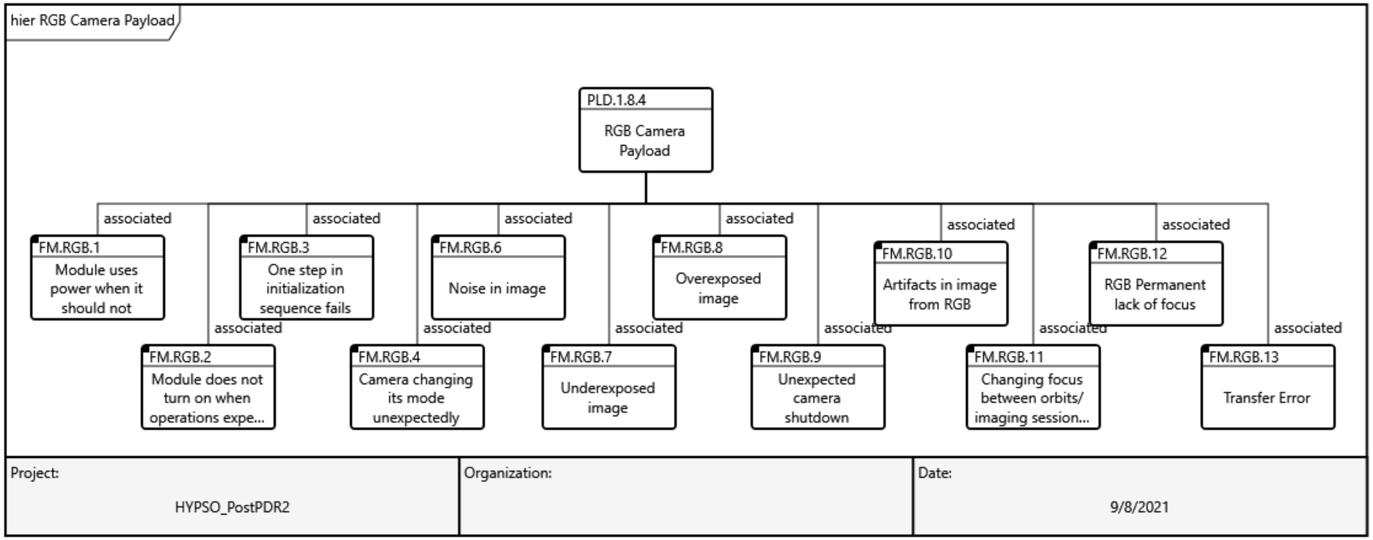


Figure B.12.: RGB camera failure modes.

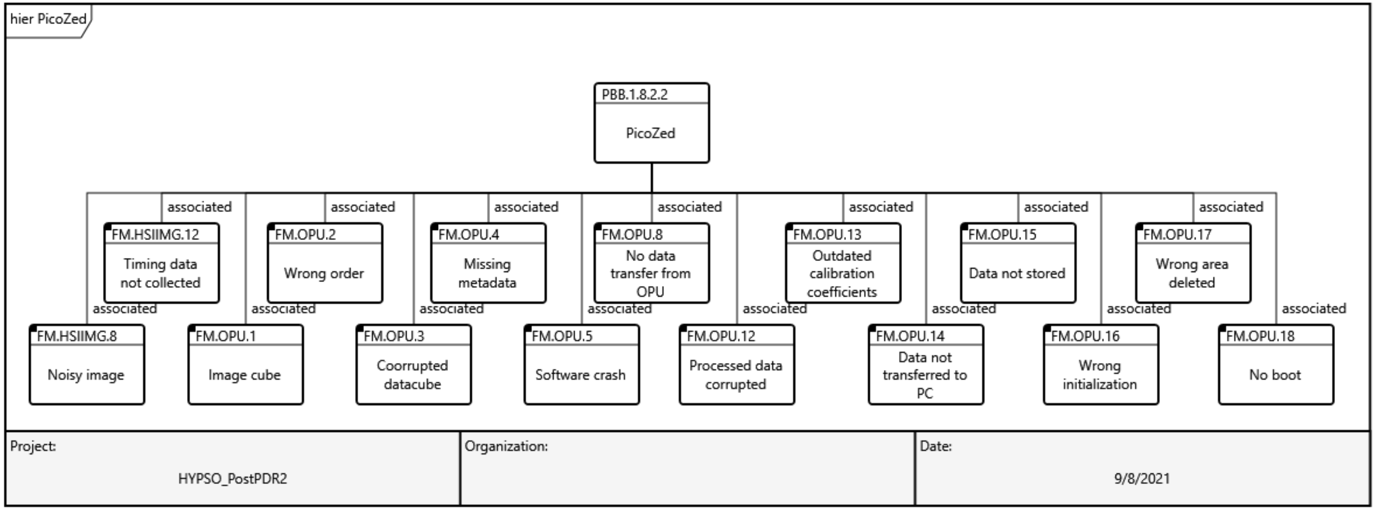


Figure B.13.: On-board processing unit failure modes.

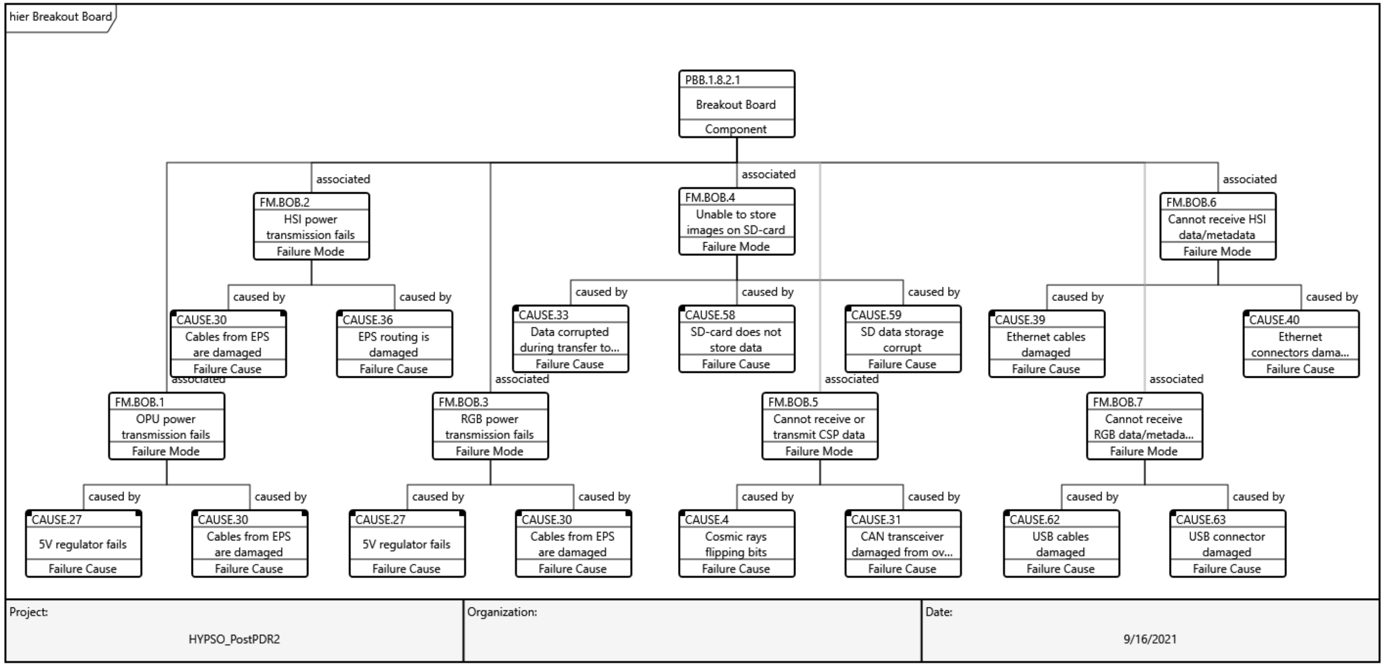


Figure B.14.: Breakout board failure modes and causes.

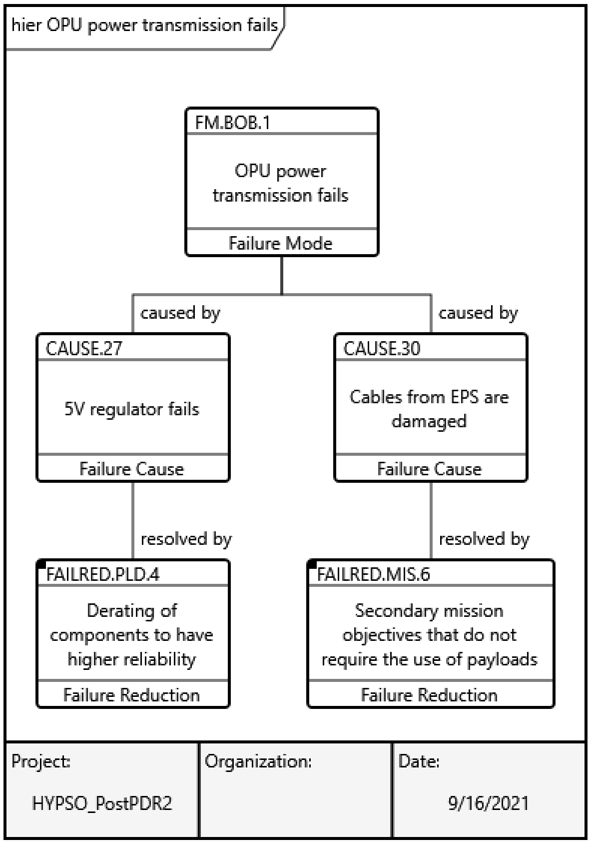


Figure B.15.: Example of failure mode, causes, and failure reduction measures for one of the BOB failure modes.

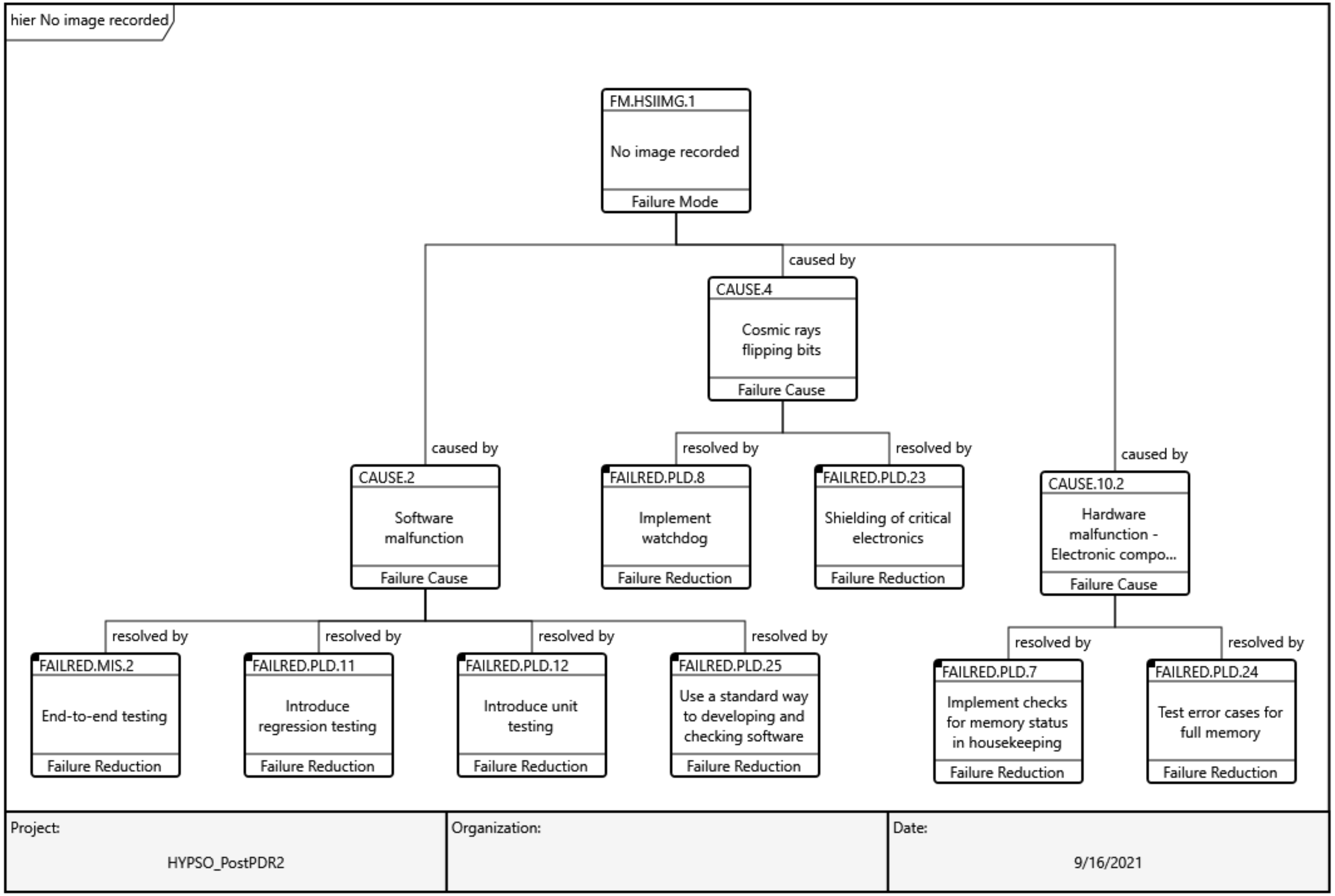


Figure B.16.: Failure mode: no image recorded example with causes and reduction measures.

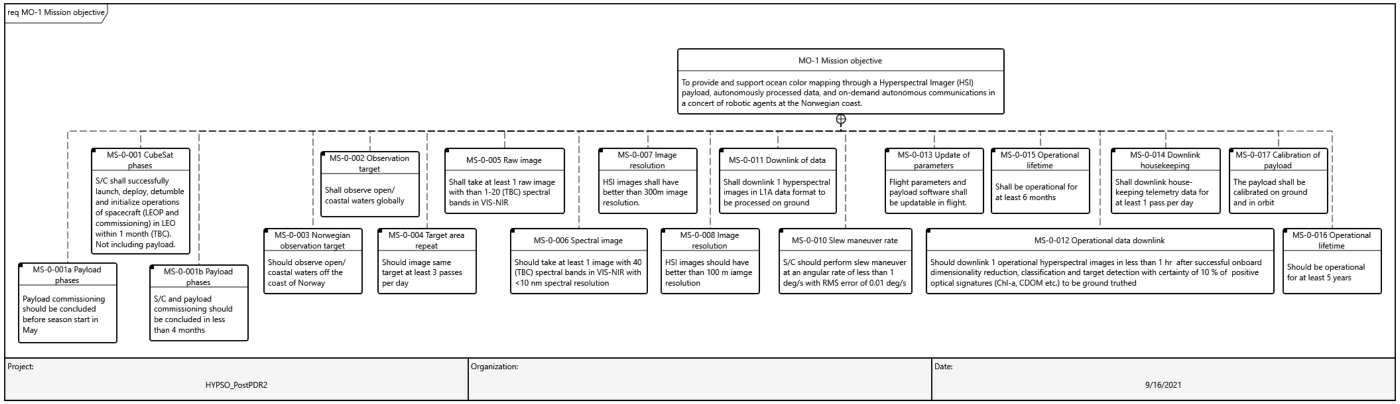


Figure B.17.: Mission objective and mission success criteria for HYPSO-1 mission as requirements diagram.

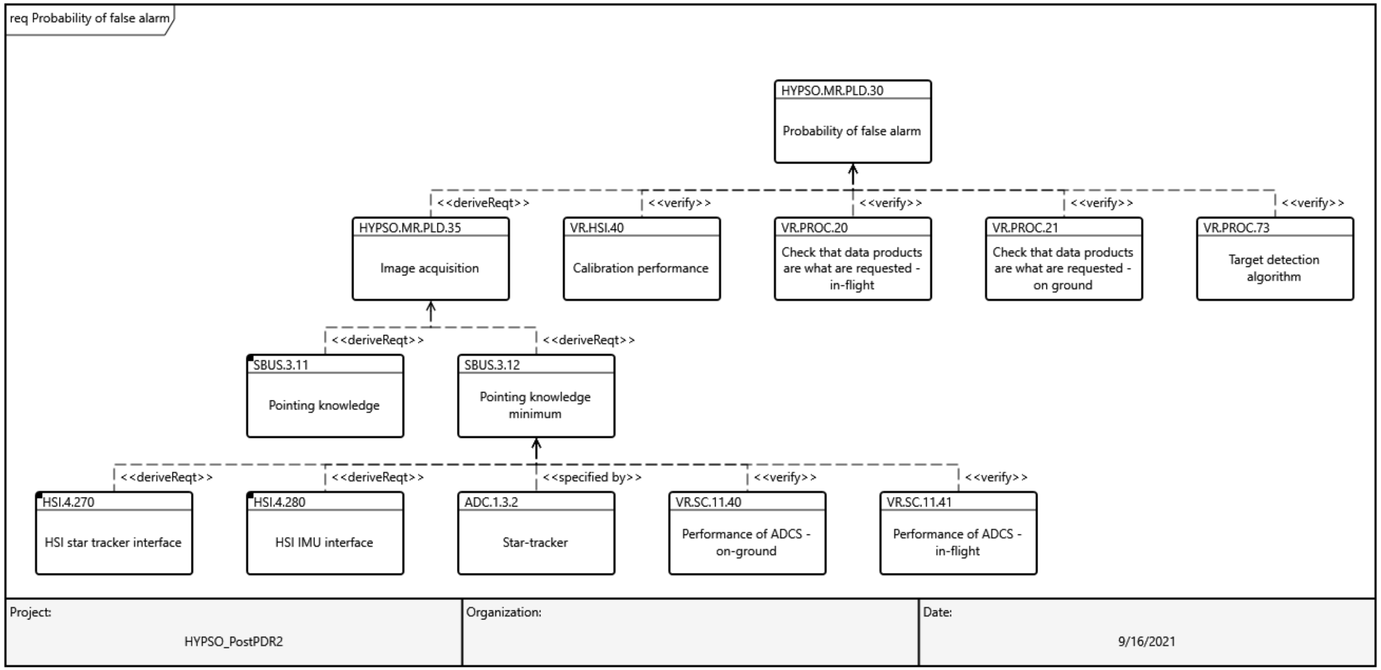


Figure B.18.: Example of requirements' breakdown.

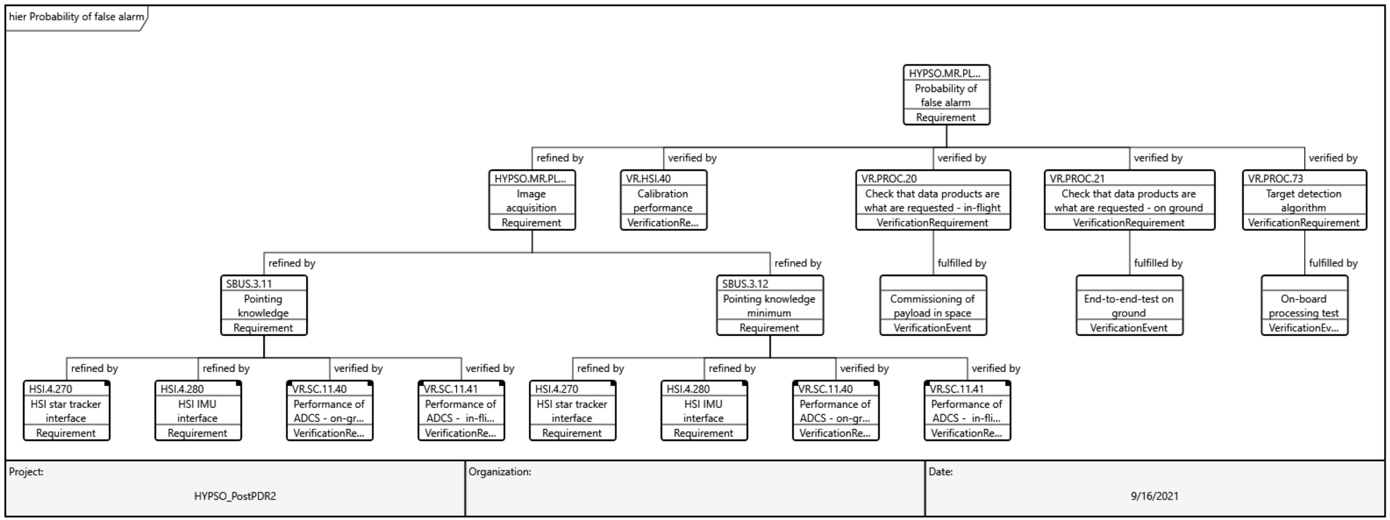


Figure B.19.: Example of verification tree with Requirement, refined requirements, and verification requirements.

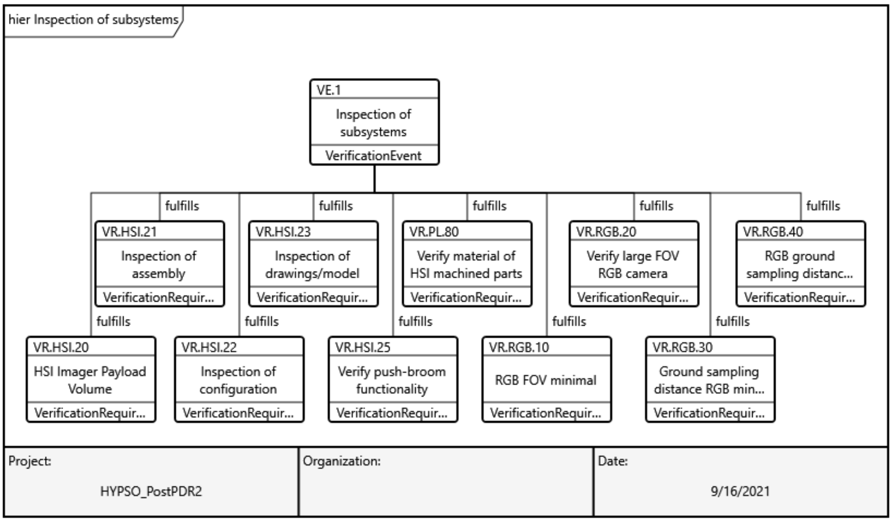


Figure B.20.: Example of verification event traceability.

APPENDIX C. N2 Diagram

Satellite Communication; Link Budget; SDR	2									TE			WE	WE								D;						
Project Manager; Schedule; SysEng;		3							PL;				SP	SP	PL;													
FlatSat; Ground Station; Satellite Bus;	ICD;		4												MA;							D; SW; ICD;						
Robotic Agents				5			D;	PL; API;	D;																			
V4/V6 Usage					6																							
Optical Correction Methods						7	SW;	ICD;	PL; SW; ICD;														D; SWA;					
Optical Characterization/Calibration; Remote Sensing; Optical Testing				D;			8		PL; D;												D;		OP;					
Mission Design; Operations Design; ADCS; Mission Requirements				D; PL;				9		TE					SP;								D; SWA;					
Target Detection; Payload SW Planning				D;				SWA;	10										SWA;		D; SWA;	SW-PL		SWA;				
Requirements Mgmt.; Mechanical Design and Analysis; Mass Budget										11	MA	MA	WE	WE	MA;									EL;				
Thermal Control System Design and Analysis; Thermal Budget; Tools										WE; TE	12				MA;								ICD;					
Payload Mechanical Interface; Satellite Bus Configuration; CoG/MoI; MBSE										WE		13	ICD	ICD											ICD			
Shock testing; Implementation		PL								ME	ME	ME	14				EL							EL; WE				
Shock testing		PL								ME	ME	ME		15			EL							EL; WE				
Outgassing and materials testing		PL;	MA;							MA;	MA;	MA;			16													MA
Adaptive Sampling									PL; TO;							17												
Geo-referencing		PL;							PL;	WE; TE;			WE	WE			18						D;					
Optical Calibration/Corrections						D; TO;			PL;	TE								19										
Data Compression; FPGA Implementation									PL;										20		SOC; SWA	D;					SW;	
Camera Communication On-board									PL;											21								
Target Detection Algorithm Implementation in FPGA									PL;										SOC		22			SOC				
CubeSat Space Protocol	ICD;		ICD;						PL;								ICD;	ICD;				23						
Image Registration; Geometric Correction				D; SWA;				D;	PL; D;								D;		D;				24				D;	
Make Payload SW Work									PL;	TE; EL; ICD;			WE	WE	MA;						SOC	SW;		25				
Nonlinear Control Algorithm for CubeSat																							D; SWA;		26			
SDR Ground Equipment																										27		
Implementation of Super-resolution (FPGA ++)						SW;			PL; SWA; ICD;												D;	ICD; SWA;					28	
Camera design						OP;	OP;			WE; TE					OP;			OP;										29

Figure C.1.: N2-diagram overview of team members for Paper A. Mechanical - ME, Optical - OP, Software - SW, Environment-space - SP, Weight - WE, Placement - ICD, Materials - MA, Tools - TO, Temperature - TE, Planning - PL, Electrical - EL, Data - D, SW Architecture - SWA, System-on-Chip - SoC.

ISBN 978-82-326-6888-5 (printed ver.)
ISBN 978-82-326-6180-0 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU

Norwegian University of
Science and Technology