Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Jenny Almestad

Facilitating for Improvement of Teamwork Skills in Software Engineering Education

An Exploratory Case Study on How Facilitators in Software Engineering Education can Contribute to Improve Teamwork Skills In Project-Based Learning

Master's thesis in Computer Science
Supervisor: Torgeir Dingsøyr
June 2022

NTNU
Kunnskap for en bedre verden

Jenny Almestad

Facilitating for Improvement of Teamwork Skills in Software Engineering Education

An Exploratory Case Study on How Facilitators in Software Engineering Education can Contribute to Improve Teamwork Skills In Project-Based Learning

Master's thesis in Computer Science
Supervisor: Torgeir Dingsøyr
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

A continuous challenge in today's educational system is closing the gap between what is taught in software engineering education and what employers look for when they hire. Students need to develop the skills and knowledge that employers seek when hiring and recruiting software engineers. In the rapidly changing and competitive software industry, teamwork skills have become a more than just nice to have. Earlier literature has emphasized the need for some scaffolding from facilitators to be able to develop proper teamwork skills. Ever since the Swedish based company *Spotify* had success with the incorporation of agile coaches the interest in the role has only increased. Despite it being a role of interest, there is a lack of empirical knowledge about agile coaching and facilitation in software engineering education. This study aims to investigate how facilitators in software engineering education can contribute to improve students' teamwork skills through project-based learning. The thesis is based on an exploratory case study were the results have been analyzed by a qualitative approach. The results in this master thesis are based on observations and interviews conducted during an undergraduate software engineering course at the Norwegian University of Science and Technology (NTNU). Students were put in interdisciplinary teams of 6-8 people and were required to participate in an agile software development project using Scrum and XP, with various team-based deliverables, presentations, and demos. Based on the collected material, this thesis concludes that by becoming more involved and aware of the challenges teams face during the development process, the facilitator can contribute to improve teamwork skills such as communication, collaboration, and leadership.

Sammendrag

En kontinuerlig utfordring i dagens utdanningssystem er å minske gapet mellom det som undervises i dataingeniørutdanning, og det arbeidsgivere ser etter når de ansetter. Studenter må utvikle ferdighetene og kunnskapene som arbeidsgivere søker når de ansetter, og rekrutterer dataingeniører. I den raskt skiftende og konkuransedyktige programvareindustrien har teamarbeidsferdigheter blitt nødvendige å besitte. Tidligere litteratur har understreket behovet for tilrettelegging fra veiledere for å kunne utvikle riktige teamarbeidsferdigheter. Helt siden det svenske selskapet *Spotify* hadde suksess med inkorporeringen av smidige coach'er, har interessen for rollen bare økt. Til tross for at det er en rolle av interesse, er det mangel på empirisk bevis om smidig coaching og veiledning i dataingeniørutdanning. Denne studien undersøker hvordan veiledere i dataingeniørutdanningen kan bidra til å forbedre studentenes teamarbeidsferdigheter gjennom prosjektbastert læring. Oppgaven er basert på et casestudie og resultatene har blitt analysert etter en kvalitativ metode. Resultatene i denne masteroppgaven er basert på observasjoner og intervjuer som er gjennomført underveis i et programvareutviklings kurs ved Norges teknisk-naturvitenskapelige universitet (NTNU). Studentene ble satt i tverrfaglige team på 6-8 personer og deltok i et smidig programvareutviklingsprosjekt ved bruk av Scrum og XP, med ulike teambaserte leveranser, presentasjoner og demoer. Basert på det samlede materialet konkluderer denne studien med at ved å bli mer involvert og bevisst på utfordringene som et team møter under utviklingsprosessen, kan veiledere bidra med å forbedre teamarbeidsferdigheter, som kommunikasjon, samarbeid og ledelse.

Preface

This research paper is written as my master thesis for the Department of Computer Science at the Norwegian University of Science and Technology (NTNU). Specializing in software development I have experienced a lot of teamwork through project-based learning courses. From this experience I have found an interest in fostering good teamwork skills and team experiences. As soon to be graduated, my interest in closing the gap between what is taught in software engineering education and what is needed in the working life also increased. With my interest for teamwork development in software engineering education in mind, I came in contact with my supervisor *Torgeir Dingsøyr* who has provided me with a great amount of support through both my specialization project and master project.

I hope that by contributing to share knowledge on the specific topics of this research paper, that it will be of value for other researchers.

Acknowledgements

I would like to thank my supervisor, Professor Torgeir Dingsøyr, at the Department of Computer Science at NTNU. Torgeir has guided me through both the specialization project and the master thesis. During this time he has offered support and guidance, answered questions, discussed challenges and solutions. Torgeir has been extremely helpful throughout the whole process. Throughout this thesis, I have obtained articles, books, and information regarding both the topic and the strategy. In addition, he has enabled me to get in contact with informants of the investigated case. Without these informants, this case study would not have been feasible within the timeframe allowed for this thesis. Torgeir is an outstanding supervisor at the Department of Computer Science, and I highly recommend him to any upcoming students.

I would like to extend a special thank you to all informants who participated in this project. My gratitude also extends to Henrik Forbord and Mathea Godø Hildre for their support and guidance during our weekly colloquia and to my friends and family for their encouragement and support.

Table of Contents

List of Figures

List of Tables

1 Introduction

This section explains the motivation and background for this research and master thesis. An explanation of the context behind the thesis, research question, contributions, scope, and limitations is provided. Furthermore, the target audience for this thesis is presented as well as an overview of the thesis structure.

1.1 Motivation

The literature review on how to improve teamwork skills in software engineering education, which was part of the specialization project, found that using project-based learning with agile methodologies can contribute to improving students' teamwork skills (Almestad and Olssen, 2021). Additionally, the study showed that through project-based learning, students of software engineering were proven to gain valuable real-world experience and authentic learning experiences. Project-based learning had also provided students with both technical and soft skills that prepared them to be good employees in the future.

The literature review revealed that a challenge students face in the current educational system is developing the skills and knowledge that employers seek when hiring and recruiting software developers (Almestad and Olssen, 2021). Employers look for applicants who possess both hard and soft skills (Vogler, 2018). A college graduate with the soft skills employers desire, includes flexibility, self-discipline, a positive attitude, willingness to learn, and teamwork skills. In order to succeed in a team, these soft skills are necessary to possess (Vogler, 2018). Having teamwork skills has become more than just something nice to have in the software development industry due to competition. The need for technical professionals to not only work in their respective disciplines, but also contribute to the business's mission, has made these skills indispensable. (Almestad and Olssen, 2021)

The technology industry looks for candidates with the "full package" when hiring graduates. They value soft skills just as much as technical skills (Vogler, 2018). Several software engineering educational programs today use agile methodologies for project-based learning. It has long been recognized that project-based learning is the best method to equip students with both hard (technical) skills and soft skills. Through this approach, students will solve a complex challenge through a real-world project. As a result, they will gain a great deal of insight into the projects they will face in the future. (Vicente, 2020)

To ensure successful use and implementation of agile methods in software development, agile coaching has emerged (Tkalich, 2020). Additionally, agile coaches address issues within the field of organisational psychology, such as team interaction. By facilitating coordination mechanisms within a team, agile coaching has shown to have the potential to improve teamwork (Bäcklander, 2019).

Ever since the Swedish based company *Spotify* had success with the incorporation of agile coaches the interest in the role has only increased (Bäcklander, 2019). Despite it being a role of interest, there is a lack of empirical knowledge about agile coaching, thus it being an attractive research topic. The existing literature also seems to lack a theoretical understanding of how agile coaching affects teams and organizations.

This brings me to the motivation for this master thesis. My interest in teamwork and improvement of teamwork skills started when I chose my specialization within Software Development. Through this specialization, I have taken several project-based learning courses, which have provided me with valuable teamwork experiences. As I prepare myself for the working world I also see the importance of possessing teamwork skills not only to be able to utilize my technical skills to their fullest potential, but also to be a good employee that others thrive around.

The purpose of this study is to investigate how facilitators in software engineering education can contribute to improve students' teamwork skills. The scope of this thesis is limited to an investigation of undergraduate students taking part in a project-based course. The goal is to investigate how students improve teamwork skills through project-based learning and how facilitators can enable this improvement to happen. This study will contribute with empirical evidence on development of teamwork skills in software engineering education.

1.2 Context

This master thesis is written at the Department of Computer Science at the Norwegian University of Science and Technology (NTNU). It is a continuation of the specialization project and literature review by Almestad and Olssen, with the focus on how to develop teamwork skills in software engineering education using project-based learning and agile methodologies (Almestad and Olssen, 2021). This master thesis is an exploratory case study investigating how facilitators in software engineering education can contribute to improve students' teamwork skills through project-based learning.

I have chosen to observe and interview university students and teaching assistants at

NTNU enrolled in the course *TDT4140 Software Engineering*, as this is a course that uses project-based learning as well as it includes interdisciplinary teamwork. Previous research has revealed that project-based learning and the nature of interdisciplinary teams can contribute to developing important soft and hard skills. However, the incorporation of both project-based learning and interdisciplinary teams does not make development of teamwork skills self-evident, as it relies on the scaffolding from the facilitator (Vogler, 2018). This research might therefore be useful in future forming of project-based learning courses in software engineering education. It can be utilized to get insight into what aspects are important when wanting to improve development of teamwork skills.

1.3 Research Question

There is a continuous challenge in educational institutions of closing the gap between what is taught in software engineering education and what employers look for when they hire (Bancino and Zevalkink, 2007). When it comes to getting prepared for the working world, keeping up with the rapid changes in the software industry is essential. In today's work environment, having good teamwork skills and the ability to work well in teams is particularly important. Technical skills are usually easier to test and educate, so the development of important soft skills is often overshadowed by this. Graduating students of software engineering are expected to start working straight away and to be able to contribute fully to their development team. Thus, it is essential for students to develop not only technical skills, but also teamwork skills in preparation for their future careers (Almestad and Olssen, 2021).

The concept of project-based learning has been recognized as a best-practice for developing both hard and soft skills in students (Gutica, 2018; Bancino and Zevalkink, 2007; Vogler, 2018; Almestad and Olssen, 2021). While studying software engineering, I have had my most positive educational experiences during projects-based learning courses. For this master thesis I investigated how facilitators can contribute to improve students' teamwork skills. I have observed and interviewed students and facilitators in the NTNU course *TDT4140 Software Engineering* of the spring 2022 semester to investigate how facilitators can contribute to improving students' teamwork skills in software engineering education. My research question is therefore:

How can facilitators contribute to improve teamwork skills through facilitating project-based learning in software engineering education?

1.4 Thesis Contributions

The main goal of this master thesis is to investigate how facilitators can improve students' teamwork skills in software engineering education. This research provides empirical evidence on how facilitators can contribute to improve teamwork skills. Qualitative data has been collected through a project-based software engineering course taught at NTNU. The results of the data collection is discussed in regard to the research question stated in Section 1.3.

1.5 Scope and Limitations

The scope of this master thesis is limited to discussion around how facilitators in *TDT4140 Software Engineering* can contribute to improve students' teamwork skills. Only findings discovered during the course are presented and discussed, as an extensive discussion of every factor contributing to improvement in teamwork skills is out of scope for this thesis. Time limitations have restricted the investigation to the spring of 2022. Discussions are limited to the data collected during the project scope. The results that are presented will be limited to the collected data, with other findings outside the scope of the research question being omitted.

1.6 Target Audience

This master thesis explores how facilitators in software engineering education can contribute to improve students' teamwork skills through project-based learning, and can therefore be of interest to course staff or practitioners within software engineering education. Similarly, researchers or students who are currently doing research or preparing reports on the same topic will also be a part of the target audience.

1.7 Thesis Structure

Table 1 gives an overview of the thesis structure and a short description of each section.

Table 1: Thesis structure

	Section	Description
1	Introduction	An introduction to the thesis which presents the motivation and background behind the research. The research question is presented as well as the contributions, scope and limitations and target audience.
2	Theory	Background literature on topics and concepts that are needed to understand this thesis' case, results, and discussion is presented in this section. This includes literature on Software Engineering Education, Scrum, Scrum with XP, Project-Based Learning, Teamwork and Facilitation Through Agile Coaching.
3	Research Method	This section presents the research strategy and corresponding research method that was used and how it was conducted. An overview of how data was collected and an evaluation of the research process is provided. Additionally the case is briefly presented as well as an explanation of the selection of participants.
4	Case	This section presents the investigated case with background information and case description.
5	Results	This section presents the results of this case study. Findings from the conducted interviews and observations will be provided.
6	Discussion	In this section a discussion of the results in relation to the literature is presented. The goal is to answer the presented research question of this thesis.
7	Conclusion	This section provides a final conclusion to the research question of this thesis, as well as suggestions for future research.

2 Theory

In this section, relevant theory and background literature is presented. The purpose of providing this background information is to introduce the topics and concepts that are needed to understand this thesis' case, results, and discussion. This section will begin with an introduction to software engineering education, followed by an explanation of the Scrum framework and how it is combined with Extreme Programming (XP) in order to provide insight into the framework used in the investigated case. Following this, project-based learning and teamwork will be discussed. Lastly, team facilitation through agile coaching will be presented.

2.1 Software Engineering Education

Software engineering education has the responsibility of preparing software engineering students with the skills needed to fulfill the expectations of the software industry (Mishra and Mishra, 2012). Thus, universities should adapt their software engineering curriculum to industry needs in order to produce highly skilled professionals (Mishra and Mishra, 2012). A challenge in software engineering education is the inability to provide students with "real-world" and large-scale software development experiences in an academic environment (Su et al., 2007). While students can have a high level of theoretical knowledge, they often lack the practice of solving real-life industrial problems (Mishra and Mishra, 2012). In order to meet the needs of the software industry, Loftus et al. suggests designing the curriculum based on software industry requirements (Loftus et al., 2011). However, this can be challenging due to the rapidly changing and growing nature of software engineering (Mishra and Yazici, 2011; Cico et al., 2021). To provide students with realistic experiences to practice their skills, many courses in software engineering education are design to rely on teaching strategies such as project-based learning, team-based learning and studio-based learning (Cico et al., 2021).

Software engineering differs from other more traditional fields of engineering in several ways, and therefore, these differences must be taken into account when teaching it (Ghezzi and Mandrioli, 2006). Lack of well-established models and notations, and less mature theoretical foundations that are not directly applicable are some of the factors that influence how practitioners teach software engineering (Ghezzi and Mandrioli, 2006). Ghezzi and Mandrioli also argue that more emphasis is needed on interdisciplinary culture and communication skills than in other engineering fields.

Further support for this assertion is made by Cico et al. who claims that agile software projects require students to communicate effectively with customers and to work with developers at different geographical locations (Cico et al., 2021). The ability to work in an interdisciplinary environment is essential for software engineers (Warr and West, 2020). It is imperative that they are able to understand problems and models from other study fields, and to interact with specialists in these fields (Ghezzi and Mandrioli, 2006). The above differences can act as an explanation for there being a wider gap between what is taught in education and what is required of the software industry in software engineering compared to other traditional fields of engineering.

Teams and individuals often work under constant pressure to deliver customer value and may disregard or under-engage in core agile practices related to reflection and learning. Hence, knowledge sharing, learning, and reflection are typically diminished (Babb et al., 2013a; Babb et al., 2013b). Failing to engage in reflective dialogues about their practices, can result in teams suffering from process erosion (Coleman and O'Connor, 2008). Agile software methods emphasize reflection as a means of learning, but their practice does not explicitly state how to create a culture that facilitates ongoing learning and reflection (Babb et al., 2014). By recognizing the learning opportunities in everyday agile practices, Babb et al. suggests that practitioners can gain a significant understanding of the domain, required technologies, and individual and team capabilities (Babb et al., 2014).

2.2 Scrum

The *Scrum Guide* (J.Sutherland, 2020), defines Scrum as the following: *"A lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems."*. As one of the most leading frameworks in software development, Scrum is used by teams to develop, deliver and sustain complex products (J.Sutherland, 2020). In contrast to other agile frameworks, Scrum focuses on getting work done, instead of focusing on values and principles. In software engineering education Scrum is the most used and common methodological approach (Cico et al., 2021). In education, Scrum is primarily used in project-based learning, relying on different student teams (Uskov et al., 2016; Wallace et al., 2012). Other teaching approaches usually utilizing the Scrum framework are customer, innovation and capstone driven courses.

An important part of Scrum is continuous delivery to customers (J.Sutherland, 2020). With Scrum, changes in requirements and goals are embraced. A product

is built by a Scrum Team in a series of sprints which includes several events and artifacts. Figure 1 gives an overview of The Scrum process, illustrating the team, artifacts and events of Scrum.

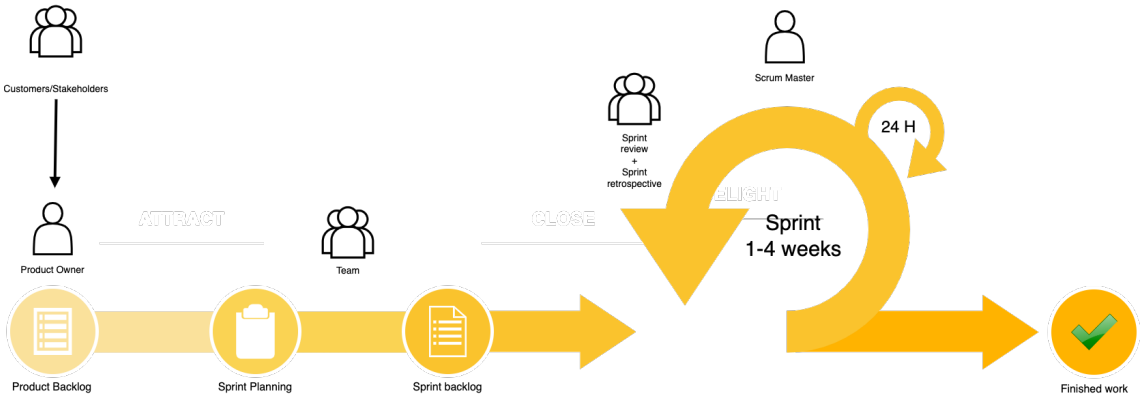


Figure 1: The Scrum process, illustrated from (J.Sutherland, 2020)

2.2.1 The Scrum Team

In Scrum the fundamental unit is the Scrum Team (J.Sutherland, 2020). This is a small team of people consisting of developers, a Product Owner and a Scrum Master.

The Developers

During each iteration, the developers continuously develop and test the product. At the end of each iteration, they provide something of value, a new version of the final product. It's up to each team to decide how to distribute the work.(J.Sutherland, 2020)

The Product Owner

The product owner is a person responsible for maximizing the value of the final product. How this is accomplished may vary from one organization, Scrum team or individual to another. Product owners are responsible for managing the product backlog as one of their main duties. Any changes to the product backlog must be agreed upon with the product owner, who represents the stakeholders' needs. Management of the product backlog consists of many tasks, these are (J.Sutherland, 2020):

- Determining the product goal and communicating it explicitly.
- The creation and communication of product backlog items.

- Ordering product backlog items.
- Making sure the product backlog is transparent, visible, and understood.

The Scrum Master

As defined in the *Scrum Guide*, the Scrum Master is responsible for establishing Scrum. This is achieved by helping everyone in the Scrum team and the organization understand Scrum theory and practice. Scrum Masters are responsible for ensuring the effectiveness of Scrum teams. In order to do this, they allow the Scrum team to improve its practices within the Scrum framework. They serve as true leaders to both the Scrum team and the organization at large. (J.Sutherland, 2020)

2.2.2 Events

All Scrum events are contained within sprints. Scrum events allow for the inspection and adaptation of Scrum items. The goal of these gatherings is to promote transparency. Failure to execute any events as directed results in missed inspection and adaptation opportunities. Scrum uses events to create regularity and reduce the requirement for meetings that aren't part of the Scrum framework. (J.Sutherland, 2020)

The Sprint

Sprints are at the core of Scrum, where ideas are converted into value. To ensure uniformity, they are usually shorter than one month, preferably 2-3 week long events. A new sprint begins soon after the preceding sprint concludes. Sprints contain all of the work required to reach the product goal, such as sprint planning, daily scrums, sprint review, and sprint retrospective. (J.Sutherland, 2020)

Sprint Planning

The sprint planning is a meeting at the beginning of each sprint, where the team makes a plan for work is to be performed during the sprint. As part of the sprint planning, tasks from the product backlog are chosen and put in the sprint backlog. The Scrum team participates in discussions and dialog to define a sprint goal and prioritize which tasks from the product backlog are to be included in the sprint backlog. The sprint backlog is an artifact made as a result of the sprint planning meeting. (Almestad and Olssen, 2021)

Daily Scrum

Daily Scrum is a 15 minute meeting which is held at the same place and time every day during the sprint. The Scrum team can structure the meeting as they wish as long as the focus during the meeting is the progress toward the sprint goal. The daily Scrum should produce an actionable plan for the upcoming day of work. This contributes to maintaining focus in the team as well as it improves self-management. (J.Sutherland, 2020)

Sprint Review

The sprint review takes place at the end of every sprint. This is a meeting where the Scrum team presents the outcome of the sprint and future plans to key stakeholders. A demonstration of the product is given by the team to the stakeholders which in return will provide feedback on the recent increment. Feedback obtained from stakeholders will be put in the product backlog and reviewed at the next sprint planning meeting. (Almestad and Olssen, 2021)

Sprint Retrospective

Sprint retrospectives are held to plan ways to increase quality and efficiency. This is a meeting where the Scrum team discusses and shares experiences from the previous sprint. A crucial component of this meeting is continuous improvement. To make future sprints even better, the team must discuss points of improvements as well as elements they benefit from and should continue practicing. One or more action points are usually defined, which are actions they will have a special focus on for the next upcoming sprint. (J.Sutherland, 2020)

Research shows that executing successful retrospectives are difficult despite the basic concept and setup of retrospectives being intuitive (Matthies et al., 2019). In the research conducted by Matthies et al., process facilitators and coaches identified the following challenges that frequently arise during retrospectives and hinder teams from realizing their retrospective's full potential: No Preparation, Not Speaking Up, All Talk–No Action and Too Repetitive. Process facilitators were then provided with activities to counteract the identified issues their team had experienced during the retrospective meetings. The research indicated that six of the retrospective activities were effective in solving the identified problems. The study showed how retrospective effectiveness and teamwork were improved by increasing awareness of how retrospective problems can be solved by using already known activities. (Matthies et al., 2019).

2.2.3 Artifacts

The Scrum framework includes a collection of recommended artifacts. These artifacts are used to maintain track of the development process, including what work has been completed and what work remains to be completed. They are intended to maximize essential information transparency. As a result, everyone assessing them has the same foundation for adaptation. Each artifact contains a commitment to make sure it delivers information that improves transparency and focus, against which progress can be monitored. This commitment is the product goal for the product backlog, the sprint goal for the sprint backlog and "definition of done" for the increment. (J.Sutherland, 2020) (Almestad and Olssen, 2021)

Product Backlog

The product backlog is a result of customer needs and a vision on how to solve it. It is a concretized vision made into a prioritized list of requirements, stories or features. These items are described using the customers terminology and usually include an ID, description, importance, initial estimate and a "how to demo" describing how the story will be demonstrated at the sprint demo. (Kniberg, 2015)

Sprint Backlog

After the sprint planning and before the first daily Scrum, the sprint backlog is created. The sprint backlog includes the sprint goal, the collection of product backlog items chosen for the sprint, and an executable strategy for delivering the increment. The sprint backlog is a strategy created by and for developers. It is a highly visible, real-time representation of the work that the developers intend to complete during the sprint to meet the sprint goal. As new information becomes available, the sprint backlog is updated throughout the sprint. It should be detailed enough for them to be able to track their progress in the daily Scrum. (Kniberg, 2015) (J.Sutherland, 2020)

Increment

The product increment at the end of a sprint demonstrates what has been finished in terms of the backlog items chosen for development and whether or not they meet the team's "definition of done". In other words, the product increment represents how far development has come. The "definition of done" says something about the criteria that must be met before anyone can say that a task completed. In teamwork where frequent partial deliveries are under development, discussions and disagreements can often arise about when functionality, code or product is finished.

"Definition of done" is used as a defined checklist to check if a task is completed. (J.Sutherland, 2020) (Almestad and Olssen, 2021)

2.3 Scrum with Extreme Programming (XP)

Scrum and XP are natural matches and can be easily combined (Kniberg, 2015). With Scrum, the focus is predominantly on management and organizational practices; whereas with XP, the focus is mainly on programming practices. As Scrum and XP are informal in nature, they are easy to adopt. With Scrum and XP, the emphasis is on producing software, thus minimizing the overhead. XP emerged in response to the overall increase in emphasis on processes, methods, and documents that the older prominent methodologies had developed. These older waterfall-based methodologies almost made software development seem like everything other than software development. (Blom, 2010)

In software engineering education, Scrum and XP have been successfully adopted. Patil and Neve suggest that by mixing the two, further benefits are provided to conducting software engineering practices (Patil and Neve, 2018). These benefits include improved implementation quality, efficiency and usability of the final product (Patil and Neve, 2018). In support of this, Cico et al. recommends that practitioners of software engineering continue to mix agile and lean methodologies (Cico et al., 2021). In addition to becoming increasingly popular in recent years, mixing agile and lean approaches is also reported to lead to positive learning outcomes and increased student satisfaction (Cico et al., 2021).

2.3.1 Pair Programming

In pair programming, two developers work together on the same computer, one as a controller and the other as an observer. The observer can provide instant feedback to the controller if any errors or deviations from coding standards occur. Controller and observer roles can be rotated to increase problem-solving as well as to expand the developer's understanding of the whole system (Blom, 2010). In his book on Scrum and XP, Kniberg argues that pair programming increases code quality, improves team focus and contributes to knowledge sharing within the team (Kniberg, 2015).

2.3.2 Test Driven Development (TDD)

As part of the TDD approach, the developer begins by writing tests for pieces of code which do not yet exist. These tests should be simple, and should only test a single feature at a time. In the next step, the developer will write the code required to make the test pass, in other words, he or she will implement the piece of functionality the test implied. In the end, the developer ensures that the code and tests are clean, following coding standards and following good practices. TDD has been proven to greatly improve system design. Even though it can take time to set up and run TDD effectively, the investment return is quick. (Kniberg, 2015) (Blom, 2010)

2.3.3 Incremental Design

Rather than trying to get the design right from the start, incremental design keeps it simple from the beginning and keeps improving it continuously. Continuous improvement in design follows naturally from using TDD. Customer requirements may change as they emerge or change over time, and the goal of the system may alter as well. Therefore, incremental design is particularly relevant in software engineering. (Kniberg, 2015)

2.3.4 Continuous Integration

Continuous integration involves integrating the entire system as often as possible. As a result of continuous integration, problems with the entire system can be detected sooner rather than at the end of a project where a traditionally developed system might detect them initially. While implementing continuous integration can be quite time-consuming, it pays off almost immediately and makes development more efficient. (Kniberg, 2015) (Blom, 2010)

2.4 Project-Based Learning (PBL)

PBL is an open-ended, learner-centered approach that emphasizes students' independence and collaboration by assigning a problem to be solved(Warr and West, 2020). The main features of PBL include a real-life project to promote learning, student autonomy, problem-solving using necessary resources, team collaboration,

and assessment methods, such as evaluation of the final product. Through PBL students' knowledge is shaped through a real-world project. As a result, it does not just provide content, but also facilitates twenty-first century skills such as collaboration, communication, and critical thinking while resulting in qualified products. With the use of real-life problems, PBL contributes to closing the gap between the educational system and the professional world. (Warr and West, 2020) (Baser et al., 2017) (Vicente, 2020)

According to an article on the importance of soft skills for hard-core technical professionals, integrating real-world projects into education can assist students that are more linear-thinking and task-oriented in their personalities to develop their soft skills (Bancino and Zevalkink, 2007). Similar results were presented in an article on improving students engagement in software development projects (Gutica, 2018). Students who had taken part in a course embedding PBL with agile methodologies reported an improvement in their communication skills, technical skills in addition to their engagement and teamwork during the course (Gutica, 2018).

2.4.1 Increased Student Motivation

Globally, PBL is recognized as an effective, reliable and realistic method of improving students' soft skills as well as increasing their motivation. The goal is to expose students to real-life challenges that are encountered by real-life clients. As a result of introducing PBL into the classroom, the teacher begins to serve as a facilitator instead of a content deliverer. Consequently, the students are free to work independently, and the teacher will only interfere or provide guidance when it is needed. By allowing students to make their own decisions on how to complete the project, they may feel more committed. Using this method has been shown to increase students' motivation to complete their projects, develop solutions, and overcome obstacles. (Holvikivi, 2016) (Vicente, 2018)

2.4.2 Interdisciplinary

To solve real-world problems, knowledge and skills from several different fields of study can be a benefit. Students are required to take part in inquiries, implement a solution and communicate with stakeholders in projects that aim to solve real world problems. It is therefore important to share and gather ideas, information and concepts from different areas of study. This will enable you to see different perspectives and gain knowledge that can lead to new thinking. Thus, interdisciplinary teams

provide the foundation for the development of good and feasible solutions. (Fisher and Newton, 2014)

In the working world, employers seek those who work well in interdisciplinary teams (Warr and West, 2020). Employees who are able to do so, will have a better chance at achieving good results as well as better collaboration in their respective teams. In 2017 The Nordic Institute for Studies in Innovation, Research and Education (NIFU), conducted an employer survey with the main purpose of mapping employers' views on the relevance and quality of education among recent graduates from universities and colleges(Støren, 2019). As part of the survey, employers were asked to give a score from 1-5 on how important they thought eight different factors were when recruiting new employees. Figure 2 presents the results of this investigation. The survey revealed that the most important factor employers look at is interdisciplinary team experience, which scored the highest.

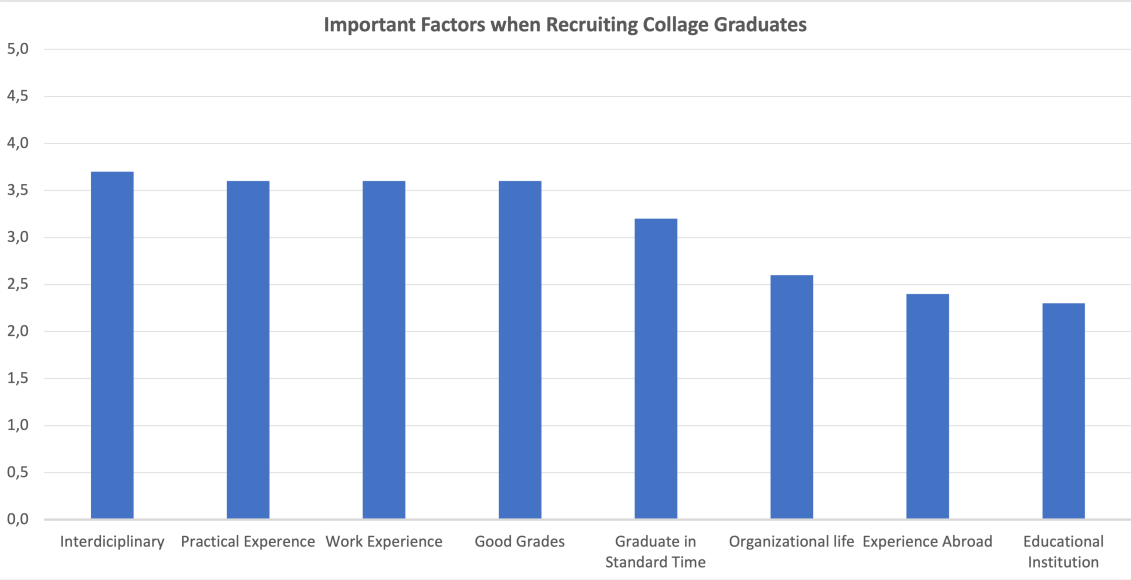


Figure 2: Average score (1-5) of how important different factors are when recruiting collage graduates. The chart is illustrated and adapted from (Støren, 2019)

In a traditional classroom environment it has been discovered to be difficult to prepare students to develop interdisciplinary teamwork skills (Vogler, 2018). In this type of environment, collaboration and the interdisciplinary aspect is missing. While working in interdisciplinary teams, students will enable development of important soft skills such as communication as they need to find a common playing ground during collaboration. (Vogler, 2018)

By giving students the opportunity to work in interdisciplinary teams, Vogler argues

that they will learn valuable experiences and develop valuable teamwork skills that will serve them well beyond their university years (Vogler, 2018). Software engineering students are often used to fill all the different roles of a development team in non-interdisciplinary projects. Whilst put in an interdisciplinary team, students are forced to assign suitable roles to team members based on their respective backgrounds and knowledge as well as being able to communicate across disciplines. In addition, utilizing both their hard (technical) and soft skills will be necessary to solve complex and real-life problems. Experience of working in an interdisciplinary team will be of great value to students when entering the working world. (Vogler, 2018)

As utilization of both hard (technical) and soft skills is a necessity in interdisciplinary teamwork, Vogler states that it does not necessarily ensure development of these skills (Vogler, 2018). To develop these skills, the team needs to be equipped with the right tools. This requires some kind of scaffolding from the facilitators. For example when met with unrealistic demands from the product owner, it is important that the team is provided with the right tools to be able to develop good and transparent communication with their product owner. (Vogler, 2018)

2.5 Teamwork

Teamwork is defined as how team members interact to accomplish a specific task or goal (Crawford and Lepine, 2013). The task and goal can be anything, for example building a house or developing an application. Teamwork is essential for employees in businesses to achieve their goals, but also to establish a pleasant work environment (Vogler, 2018). In 2017, The Nordic Institute for Studies in Innovation, Research and Education (NIFU) conducted a employers survey to map out employers' views on the relevance and quality of education among recent graduates from universities (Støren, 2019). Over 5000 businesses in both the public and private sector completed the survey. Among several questions, they were asked what they would choose if they were to choose between hiring a person with "good expert knowledge, but weaker communication and collaboration skills", or vice versa. Possibly one that is moderately good in both. 55% answered that they prefer the one who was best at communicating and collaborating. Their reasoning was that collaborative skills are harder to learn than expert knowledge (Støren, 2019).

Due to the complexity and globalized nature of our economic environment, collaboration between people, teams, departments, organizations, and industries is becoming more prevalent (Xia et al., 2020). Employees are required to work effectively in

teams, and possess teamwork skills. In order to meet the demands of the 21st century workplace, it is therefore important that engineering programs in universities provide experiences to develop students' teamwork skills sufficiently.

Teams with members who lack basic teamwork skills can be disruptive to the team process as well as its productivity (Largent, 2016). In an article by Largent, on measuring and understanding team development he argues that team members who communicate early about a given challenge to their team are more likely to see it resolved together. The inability of a team member to communicate effectively can lead to members distrusting them because they have broken their commitments in the past. This is simply because they did not realize anything was wrong until it was too late to fix it. (Largent, 2016)

Agile software development methodologies depend heavily on teamwork. With the continuous move towards agile software development methodologies, teams capable of working effectively together have become increasingly valuable. Due to the methodology's heavy reliance on teamwork, individuals who are not skilled at working in a team environment can very quickly undermine the productivity of an agile team. (Chong and Hurlbutt, 2007) (Largent, 2016)

2.5.1 Teamwork Skills

Despite the fact that many students are able to acquire technical skills, less are able to communicate those skills effectively or function well in a multidisciplinary team (Cerato et al., 2012). Teamwork requires a wide range of skills (Vogler, 2018). Teamwork skills are developed during interaction and working together in a team. The skills developed through such team experiences are considered soft skills. These types of skills can not necessarily be taught, but are developed through experiences and encounters in your life. (Vogler, 2018).

In an article by Thompson on developing teamwork skills in an introductory engineering course, students reported that team success was dependant on how they distributed tasks, if everyone on the team participated equally and if they were able to complete the product within the time scope (Thompson, 2017). A variety of studies have consistently shown that cooperative learning is extremely effective at improving almost every kind of learning outcome (Yusof et al., 2012; Terenzini et al., 2001; Johnson et al., 2000). However, Oakley argues that simply putting students in teams to work on assignments is not sufficient to achieve these benefits (Oakley et al., 2004). The instructor must ensure that the teams develop the skills

associated with high-performance teams, otherwise the teams' learning experience will be ineffective and may even fail. Immediately after forming a team, Oakley et al. suggests establishing the policies that will govern their operations, and asking team members to develop their own expectations. Students may also benefit from being informed about some of the mistakes that new teams commonly make and how to avoid these mistakes. (Oakley et al., 2004)

Many skills are associated with high performance or effective teamwork. Among these we find: Collaboration, communication, time management, critical thinking, contribution, problem-solving and adapting (Xia et al., 2020). Three of the skills listed above, as well as creativity, are referred to as the 4 C's of 21st Century Skills: Collaboration, Communication, Critical thinking and Creativity. These skills are essentials in the modern day workplace and are often the most impactful when it comes to students applying and starting their careers. (Chiruguru, 2020)

Collaboration

Collaboration is a key component of teamwork. It is defined as the situation where two or more people work together to achieve the same goal. Although it can be challenging at times, it is also an excellent opportunity to create and share ideas, experiences, and improve skills. As a student, every group project you work on can serve as a learning opportunity. When you are eager to learn and open to exploring new ideas, you will become a more valuable team member. (Almestad and Olssen, 2021)

Communication

Having a clear understanding of expectations and responsibilities is a key part of teamwork. The establishment of a team environment in which everyone can freely express themselves promotes trust and a positive climate. While it may be impossible to avoid disagreements in a team, open communication can help resolve the problem quickly and hopefully prevent arguments. (Almestad and Olssen, 2021)

Critical thinking

Decisions can be made more rationally, smarter, and more informed through critical thinking. Within a team, the tendency is to simply accept whatever decision is made without second-guessing it. There is the possibility that the decision may be made solely based on one individual's opinion of what is right, ignoring other potentially better solutions or ideas. In order to think critically, you must be able to view the issue from various perspectives. In addition, you must take prior experience into account, as well as pay attention to your team members' opinions. This is how

teams can develop creative ideas and move forward. (Almestad and Olssen, 2021)

Creativity

Critical thinking and problem-solving are closely related to creativity. To develop, implement, and communicate new ideas to others effectively, creativity is required. Global competition and the automation of many tasks have led to an environment where innovative abilities and a creative spirit are expected to be key components of professional and personal success. (Almestad and Olssen, 2021)

2.6 Team Facilitation Through Agile Coaching

In order to facilitate the adoption of agile methods and the development of agile teams, we can introduce an agile coach (Daljajev et al., 2020). Since all agile methods of software development rely on teamwork and self-managing teams, coaching has become an essential part of the process. As an agile coach, you are therefore responsible for helping your team in reaching higher levels of achievement and performance (Daljajev et al., 2020). In the article by Tkalich on agile coaching in Norway and the USA, she argues that agile coaches improve teamwork in software teams (Tkalich, 2020). The same conclusion is made by Bäcklander, he implies that by facilitating coordination mechanisms within a team, agile coaches have shown to have the potential to improve teamwork (Bäcklander, 2019).

The areas agile coaches work on are quite similar to those of organisational psychologists: they improve collaboration in development teams, motivate employees, guide leaders, and promote new organizational attitudes and habits (Bäcklander, 2019). In a study conducted on software engineering students a scrum-based model was presented and enhanced with agile coaching to maximize students performance (Rodríguez et al., 2016). The study revealed that the students who received coaching experienced gained valuable insight into the integration of Scrum, problem solving, and guidance by checkup meetings in comparison to the students who did not receive coaching.

Agile teams aim to be autonomous and self-managing (Stray et al., 2018). This type of team organizing differs from traditional team structures where there is a defined leader (Hoda et al., 2013). Rather than being controlled by a single individual, autonomous teams distribute leadership among a group of individuals instead of having one individual handle all the responsibilities. In the article on autonomous agile teams, Stray et al. identifies one of the challenges that autonomous teams face as lack of coaching and organizational support (Stray et al., 2018). In teams with

limited prior experience with autonomy and self-management, Stray et al. suggest that leaders must set the direction for the team before they can coach them towards autonomy (Stray et al., 2018).

2.6.1 Team Coaching

The role of an agile coach is not to be confused with the Scrum Master (Hoda et al., 2011). Both roles strive to maximize a team's performance by adapting to agile principles, values, and practices. However, even though the roles might seem similar the agile coach is less interested with the Scrum framework and more interested in teamwork, performance and leadership (Bäcklander, 2019). With this in mind, we can introduce the term team coaching, which refers to direct interaction with a team, intending to help members of that team employ their collective efforts and resources effectively (Hackman and Wageman, 2005). In an article on understanding and improving teamwork in organizations, Salas et al. describe team coaches to be essential for teamwork since they provide support and help identify performance gaps (Salas et al., 2014).

As a component in team coaching, we have team facilitation (Hawkins, 2017). In the book "Leadership Team Coaching", Hawkins describes team facilitation as a person that helps manage the process for the team so that they can focus on their tasks (Hawkins, 2017). The role of a team facilitator is to assist a team with a process or to have a specific conversation, usually all within the context of a few interventions (Widdowson et al., 2020). By asking specific questions, assisting team members manage conflict and encouraging knowledge sharing facilitators can help open discussions (Widdowson et al., 2020).

3 Research Method

This section presents the research strategy and research method used in this thesis. Figure 3 shows the selected research path for this master thesis. The path includes a chosen research strategy, data collection methods and data analysis method. The chosen path is outlined and highlighted in blue. The purpose of this section is to explain the path and the choices made. In preparation for this master thesis, a literature review was conducted during the fall of 2021 as part of the specialization project. Thus, the literature review is not described or explained in this thesis.

Section 3.1 will present and give an explanation of the choice behind the research strategy. Further on, Section 3.2 will present a brief introduction of the case and Section 3.3 will describe which method was used to collect data, and how it was conducted. Lastly, Section 3.4 will explain the approach chosen for data analysis and Section 3.5 will provide an evaluation of the research process.

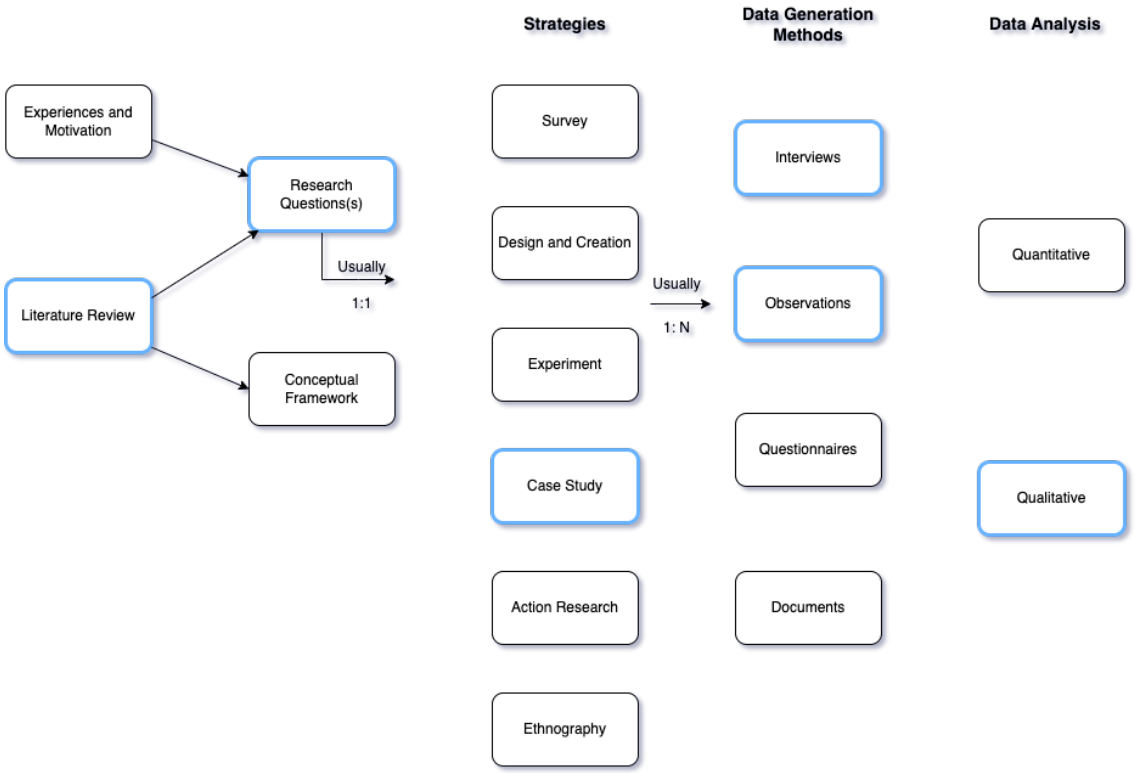


Figure 3: Model of the research process from (Oates, 2006). The strategy for this thesis is marked in blue.

3.1 Research strategy

To conduct a research study it is important to have an overall plan. This is achieved by having a research strategy. For this master thesis a case study was conducted. By selecting a strategy, the researcher is provided with helpful support when planning, executing and monitoring the study (Johannesson P., 2014).

3.1.1 Case Study

A case study is an empirical in-depth study of a person, group, organization or family. The aim is to get insight into the real-life of the particular instance and its processes and relationships (Oates, 2006). The first step in a case study is to identify and define the case that is going to be studied. Usually a search to determine what is already known about the case is conducted. This can be done through a literature review, report, media and more. The preliminary search serves to establish a basic understanding of the case and the development of the research questions that are raised. The data collected in case studies can be both qualitative and quantitative, therefore several different research methods can be used in a case study. Figure 4 illustrates the different phases of the case study process. As Robert K. Yin (Yin, 2009) describes, it is a linear but iterative process.

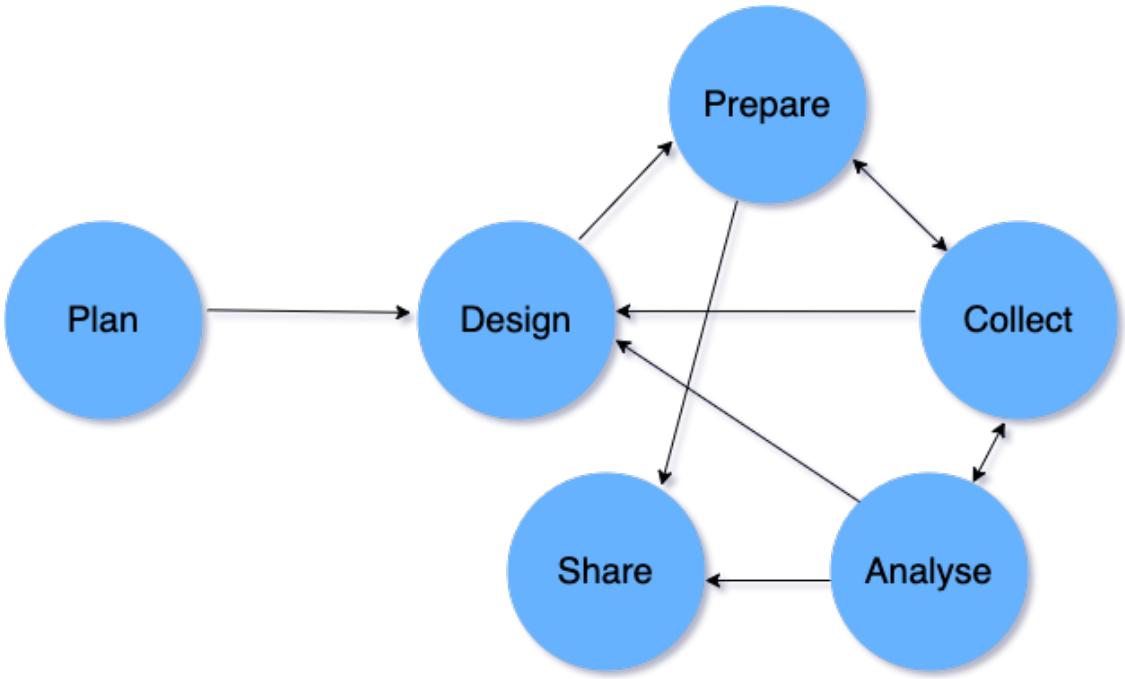


Figure 4: An illustration of the case study process from (Yin, 2009).

There are various types of case studies: exploratory, descriptive and explanatory. Exploratory studies are usually the starting point of a study and the forerunner of large-scale investigations. The results of an exploratory study are often used to suggest why and how further research should be conducted. A descriptive study is on the other hand usually used when one already has an existing hypothesis. The aim is to find a connection between the subject being studied and the suggested theory. The results will usually lead to further development of the theory. Lastly we have explanatory studies. This kind of case study is used to investigate the cause of an event. Usually an event has occurred, and the study will aim to find possible causes and predict future occurrences of such events. The results of explanatory case studies are absolute and definite, with no room for interpretation.

For this master thesis an investigation of how facilitators can contribute to improve teamwork skills in software engineering education was conducted. The investigation was conducted through a project-based learning course with interdisciplinary student teams. As part of my study, students and facilitators in the course *TDT4140 Software Engineering* at NTNU were interviewed and observed to get in-depth insight. An exploratory case study was therefore chosen as the research strategy.

3.2 Case

A brief introduction to the case will be presented in this sub-section, as well as an explanation to why the case was selected. Further background information on the case will be presented in Section 4.

The objective of the case is to investigate how facilitators can contribute to improve teamwork skills in software engineering education. To create an understanding of how facilitators can achieve this improvement, the bachelor level course *TDT4140 Software Engineering* was chosen for further investigation. *TDT4140 Software Engineering* is a course at NTNU, that uses project-based learning with interdisciplinary student teams. During the course students demonstrate that they are able to plan and manage a software engineering project using the agile methods Scrum and XP. With a facilitator and product owner at their disposition, students are required to work together in teams and reflect upon their own role. With the given course content and focus on teamwork, *TDT4140 Software Engineering* became a good fit for this particularly case study.

3.2.1 Selecting Participants

In order to gain a deeper understanding of facilitation and to answer the research question stated in Section 1.3, both students and facilitators in the course were interviewed and observed. All facilitators in *TDT4140 Software Engineering* were assigned a group (called a village) consisting of 6 facilitators. In *TDT4140 Software Engineering* their is in total four villages. For this case study one of these villages was given to the researcher for disposition. This village consisted of 6 teaching assistants and a village leader coordinating the work of the teaching assistants. The teaching assistants in the village were responsible for 18 student teams in total. Participating in the study was completely voluntary, and 5 facilitators chose to participate. The students were contacted by the facilitators who participated in this study, each of which sent out an email asking students if they would like to participate. In result, 6 students volunteered to participate. As this case desires to investigate the experiences and perspectives of facilitators and their students, it was important to interview students of the facilitators that had been interviewed. An overview of the conducted interviews can be seen in Table 2.

3.3 Data collection

To produce empirical data and evidence, we need a method to collect data. We can sort data into two different categories, qualitative or quantitative. Quantitative data is numerical, for example number of students or number of satisfied customers. Qualitative data is not numeric, but other types like words, images or sound etc. There are several different methods that can be used to generate the data. Interviews, observations, questionnaires, documents are some of the most common methods. Some of the methods are often associated with a specific research strategy, for example questionnaires with the survey strategy and observations with the experiment strategy. It is also normal to combine several methods within one research strategy. This is called triangulation. By using more than one data generation method you most likely will generate more data and the quality of research will also increase. Using several methods also allows the data to be cross-examined and questioned by comparing the data from the different methods. If the data from the different methods is consistent, it increases the quality of your findings and your research.(Oates, 2006)

For this thesis interviews and observations were chosen as the data generation method. Conducting interviews and observations produces mainly qualitative data,

which is the preferred type of data for this thesis. Qualitative data was chosen as the preferred data type as it typically involves a systematic and detailed study of individuals in natural settings and enables understanding of in-depth situations of the interviewee's perspectives and experiences (Kaplan and Maxwell, 2005).

As a teaching assistant in *TDT4140 Software Engineering*, the author of this thesis performed observations in addition to conducting interviews. By observing people, researchers find out what people actually do, rather than what they report that they do (Oates, 2006). Interviews and observations are used together to enable method triangulation. By comparing the outcomes from one method with the results of another method, the findings from one method can be affirmed or disputed, which can increase the validity of the case study (Oates, 2006).

3.3.1 Interviews

Interviews is a method commonly used in case studies. In an interview setting the researcher wants to gain information and insight from the interview subject. Conducting successful interviews requires both planning and a set of certain skills (Oates, 2006). As preparation for conducting interviews, some background information about the interviewee and the context should be gathered. In addition, it is important to be aware that the responses from the interview will depend on the perceived role and identity of the researcher. Your age, sex, accent and status will be a possible influence on the responses from the interviewees. (Oates, 2006).

There are three types of interviews: structured, semi-structured and unstructured. Structured interviews follow a strict plan and all questions are identical and standardized. It is very important that the interviewer reads out the questions the same way so that their personal views do not shine through. Semi-structured interviews, on the other hand, are not that strict and planned out. Usually the interviewer will have a list of topics and questions that they want to ask, but it is OK to go off topic and change the order of the questions depending on how the conversation goes. The last of the three types, unstructured interviews, start with the researcher introducing a certain topic and letting the interviewee talk freely about their thoughts, beliefs and knowledge. In this type of interview it is important that the interviewer does not interrupt the interviewee so that the responses are not influenced in any way. Semi-structured and unstructured interviews are more flexible than structured interviews, and let the interviewer talk more freely and off topic.

Semi-structured interviews were conducted during this case study. This type of

interview was chosen as it allows for altering and adding questions during the interview, enabling more in-depth information gathering whilst still covering the planned questions and topics and because of the need to cover specific topics the interview needed to be somewhat structured. An interview guide was created and used when conducting the interviews (Appendix A). It was beneficial to have the interview guide to ensure that all relevant topics were covered, in addition to having semi-structured interviews to have the flexibility to change a topic if the need arose. As most informants' mother tongue is Norwegian, each interview was conducted in this language. It provided all informants with the opportunity to speak more freely and explain their views more clearly.

The interviews were conducted by the author of this thesis. With a limited time scope of this case study the data collection only consists of information gathered during the course duration of *TDT4140 Software Engineering*. The interviews were carried out in two rounds. Firstly, 5 facilitators were interviewed in week 10 and 11 of 2022. During this time the facilitators had experienced several weeks of facilitation and their student groups had completed their first sprint demos. Secondly, 6 students from some of the facilitators' teams were interviewed in week 12 and 13 of 2022. In total 11 interviews were conducted. Table 2 gives an overview of the conducted interviews. The interviews were conducted and transcribed by the author of this thesis. The first round of interviews with facilitators was focused on the work of the facilitators, as well as their thoughts on teamwork, interdisciplinary teams and agile development. The second round of interviews with students focused on the involvement of the facilitators in their teamwork, the teamwork they experienced in their interdisciplinary team and their experience with Scrum.

Table 2: Overview of conducted interviews with facilitators and students in *TDT4140 Software Engineering*.

	Facilitators			**Students**	
#	**Date**	**Team**	**#**	**Date**	**Team**
F1	10.03.2022	T2, T3	S1	28.03.2022	T1
F2	16.03.2022		S2	28.03.2022	T2
F3	16.03.2022		S3	30.03.2022	T3
F4	16.03.2022		S4	30.03.2022	T3
F5	17.03.2022	T1	S5	31.03.2022	T3
			S6	31.03.2022	T2

3.3.2 Observations

The author of this thesis worked as a teaching assistant in *TDT4140 Software Engineering* the spring of 2022. As a teaching assistant you are both a facilitator and product owner for three teams and evaluator for an additional three teams. Through the job, you can observe your teams work together and how they communicate with each other through weekly meetings and descriptions from their deliverables.

As an observer, there are several types of participation: complete observer, complete participant, participant-observer and practitioner-researcher (Oates, 2006). A complete observer is present in the observed situation, but does not contribute or take part in the observed situation. The complete participant is the opposite of a complete observer. This type of participation takes full part in the situation and tries to get insight into how the situation looks from the inside. The participant-observer type shadows someone, and can be used when you lack the required credentials to fully participate in the situation. The last type of participation is the

practitioner-researcher, which is an observer that already has a job within the investigated environment, and decides to put on a researcher hat. (Oates, 2006)

For this case study the practitioner-researcher was chosen because it naturally fits in since the researcher of this study works as a teaching assistant in the course that is investigated. This type of participation might save time for the researcher, but requires some precaution. Even though the researcher already is a part of the studied environment, permission to conduct research should be obtained. It is also important to be aware of your own assumptions and pre-conceptions about your job so that issues will not be overlooked. (Oates, 2006)

The observations made during this case study included attending weekly meetings with three student teams, in addition to sprint demos and evaluation of their deliverables. None of the students that were observed were included in the conducted interviews as the power relationship between the researcher and the students could have affected the results provided.

3.4 Data analysis

The conducted interviews were transcribed and imported to QSR International's *NVivo*. By using *NVivo*, a case study database was created. *NVivo* is a software program used to analyze and organize qualitative data, and can be of good help during the research process. By using *NVivo*, a case study database was created. In his book on case study research, Robert K. Yin (Yin, 2009) encourages the creation of such a database as it increases reliability and maintains the quality of results.

For this study a qualitative analysis was conducted, based on the chosen research strategy and method. Qualitative data analysis enables a more detailed and in-depth investigation of the chosen research topics and question. Based on the approach described in Oates' book on research on information systems and computing, the data was first transcribed, key categories were identified and lastly, statements were coded.

Using the top-down approach, relevant statements were coded into nodes based on topics relevant to the research question. When analyzing a specific topic, the corresponding node was searched and looked through in the hope of finding relevant information and statements from the interviews. By doing so, the work of going through all of the conducted interviews in search of a specific theme was made a lot more effective.

3.5 Evaluation

An important part of research is to be able to be critical of your own research process. This section will evaluate the reliability and validity of the conducted research and information gathering for this master thesis.

3.5.1 Strategy and Method Limitations

A case study was chosen as the research strategy for this project. An alternative research strategy that would be interesting to choose for this study would be ethnography. Ethnography is a strategy used to explore and examine cultures and people over a longer period of time (Oates, 2006). Due to time limitations and the restricted size of this case study, ethnography was not suitable for this master thesis. Nevertheless, it would have been an interesting strategy to choose in future research with the same topics and research question.

As described in Section 3.3 a suitable data collection method was chosen for this case study. Oates argues that the use of more than one data generation method will corroborate findings and enhance their validity. This is called triangulation (Oates, 2006). For this case study, interviews and observations were the chosen data collection methods. This enables triangulation which contributes to increasing both the validity and consistency of the data generated.

Semi-structured interviews were conducted as part of this study. In an article about qualitative interviews in IS research, Myers and Newman presents several potential limitations regarding studies with semi-structured interviews (Myers and Newman, 2007). A limitation Myers and Newman describe is "lack of time". This limitation is further elaborated as the limited time for collecting empirical data and the limited time each interviewees has to spear. Due to "lack of time" the collected data might not be complete and limited as important aspect might have been left behind. In this study, limited time has affected the number of interviewees that ideally would have been interviewed. Therefore this study might not give a correct representation of how teamwork skills are developed in all software engineering courses, but can give insight and an explanation based on the chosen case.

As mentioned in Section 3.3.1 and by Oates, the quality of a interview and the data generated is dependant on the role, skill set and identity of the interviewer (Oates, 2006). As this study is conducted by a researcher with little previous experience of performing qualitative case studies, this also acts as a limitation in this study.

3.5.2 Case Study Validation

Robert K. Yin describes four tests that have been commonly used to establish the quality of any empirical social research. Since case studies are a form of an empirical study, these four tests are applicable to the case study completed in this master thesis (Yin, 2009). The following section is structured according to these four tests: construct validity, internal validity, external validity, reliability.

Construct Validity

Measures and identification of correct operational concepts are the main purposes of construct validity, based on the topics being investigated. The best way to ensure this is to use multiple sources of evidence, to establish a chain of evidence, and to have key informants review the draft. (Yin, 2009)

As preparation for this case study, a literature review was conducted as part of the specialization project (Almestad and Olssen, 2021). The literature review investigated theory and findings on the topics which this case is based on. A research question is defined to get a clear understanding of what the case investigates. Interviews and observations were the chosen data collection methods, ensuring more than one source of evidence. The chain of evidence is obtained by the interview guide and all statements presented in Section 5 can be backtracked to the transcribed interviews.

Internal Validity

As Yin describes in his book "Case Study Research", internal validity is not relevant for exploratory studies, but used for explanatory or causal studies. (Yin, 2009)

External Validity

External validity is the third test, and deals with finding out if the findings from the study are generalizable and that the selected cases are representative of the type of study (Yin, 2009). The criticism of single case studies frequently suggests that they cannot be generalized. However, Yin argues against this idea in his book and implies that such critics are implicitly comparing the situation to survey research, which relies on statistical generalization whereas case studies rely on analytical generalization (Yin, 2009). Yin suggests that using a know theory can increase external validity (Yin, 2009). This study is not based on any known theoretical model as the research in this study has a special focus on facilitation in software engineering education. During the literature review it was discovered that literature on this topic is lacking in current research, which resulted in the decision to not apply any

theoretical model to the analysis as no fitting model was found.

The informants in this case study participated voluntarily, which might indicate a bias in participants as they potentially might be more interested in teamwork than others. This study is also conducted i Norway, which in comparison to other countries might have other opinions on factors contributing to improvement of teamwork.

Reliability

The main goal of the reliability test is to minimize biases and errors in the study. We want to ensure that any future investigator would be able to reproduce the same findings and conclusions based on the same procedures and case study described by the earlier investigator (Yin, 2009). In order to increase reliability, Yin recommends that the methods and processes must be documented as well as the creation of a case study database. In this study the methods and processes are documented in Section 3 and Section 4. A case study database was also created using *NVivo* as described in Section 3.4.

3.5.3 Research Ethics

All interview participants received a letter describing the research project and informing them about their participation. Participants are informed that their identity is completely confidential and that they can withdraw from the study at any time without incurring any consequences.

The study only collects team numbers and no sensitive information. It is important to be able to identify how many different teams have been interviewed in the course to get a clear perspective of the scale of the project. Having this information can also help increase the credibility of the case study.

4 Case

This section will present the background for this case and the project that was used as a case for this thesis. Section 4.2 will present how roles are distributed among teaching assistants, and describe the role and responsibilities of the facilitator.

4.1 Background

The main purpose of this case is to investigate how facilitators can contribute to improve students' teamwork skills in software engineering education. The examined project used in this case study, is a part of the course *TDT4140 Software Engineering* held at NTNU during the spring semester of 2022. This is a project-based course, where students will experience different types of software processes, project management and planning. Throughout the course, students will have to participate in an agile software development project using Scrum and XP, with various team-based deliverables, presentations, and demos. Students are put in interdisciplinary teams of 6-8 people. The vast majority of the students who are enrolled in the course are 2nd year engineering students at the NTNU Trondheim campus. Being one of the few NTNU courses that fosters teamwork and interdisciplinary collaboration among students, this course serves as an interesting course to study for this case study.

Figure 5 shows an overview of the course and scheduled deliverables. During the first three weeks, students worked individually and were tested on topics such as agile software development, software architecture, and quality. From the fifth week on, students were put into teams. Each team was assigned a teaching assistant, which provided the team with two roles. The product owner and process facilitator. As of the fifth week, the project began. Each team was given a problem from their product owner, simulating a real-life problem. Students were going to create a product called *GroupUp*, a platform intended to facilitate new social acquaintances. Detailed product descriptions and the vision can be found in Appendix C. During the fifth week, teams were asked to create a team contract, informing each other of the team's expectations, as well as their own. Additionally, they were required to hold their first sprint planning meeting and schedule their first meeting with the product owner.

The first iteration was scheduled from week 6-9 and the second iteration from week 10-12. During the weeks of each iteration, the teams were responsible for planning and managing their time and development process. Teaching assistants scheduled

weekly meetings with their teams, in each meeting allocating time for both the role as process facilitator and product owner. The duration of these weekly meetings ranged from 20-40 minutes for each team. Distribution of time between the two roles was up to the teaching assistant. At the end of each iteration the teams were expected to conduct a sprint retrospective and demos. The end of the second iteration marked the end of the project work in the course. From then on students did no longer work in teams, but individually. The remaining weeks were used for individual reflections and peer reviews.

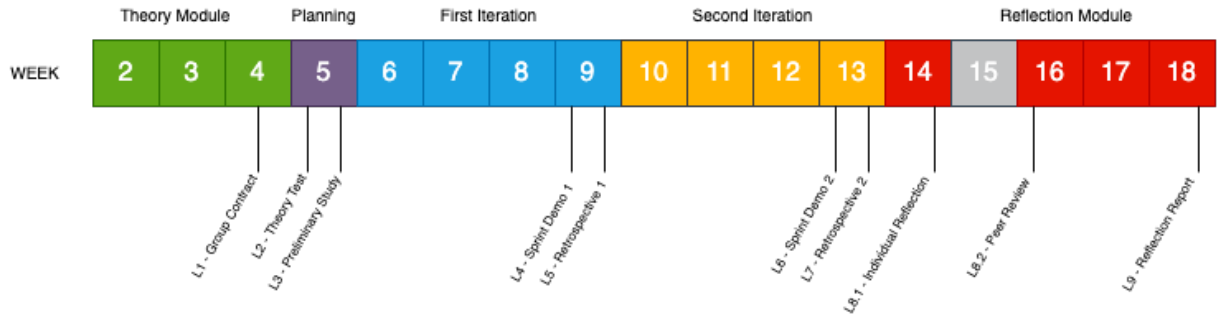


Figure 5: Overview of the course schedule and planned deliverables. Illustrated from (Farshchian et al., 2022).

4.2 Role Distribution

Three roles were assigned to teaching assistants in *TDT4140 Software Engineering*: Product owner, process facilitator, and evaluator. Figure 6 illustrates the distribution of roles among teaching assistants. The assistants were both product owners and process facilitators for three teams, and evaluators for three other teams.

Process facilitator

The goal of the facilitator was to guide the teams through the project with focus on the learning objectives, deliverables and the development process. A description of the role as a facilitator was given to all teaching assistants and can be seen in Appendix B. During their weekly meetings with their teams, the facilitator was expected to be curious and challenge the teams with questions about different aspects of their work process. This would hopefully force the teams to reflect on their teamwork and dynamics. The course staff provided all teaching assistants with a suggested agenda before every weekly meeting containing questions and topics that

might be useful to discuss with the teams (Appendix D). It was not expected that the facilitator would be able to answer technical questions, since their role was to guide the participants through the process, not to provide product development advice. Every facilitator was expected to come prepared to every meeting with an overview of what the team was working on and any obstacles they may be facing. In the role description of the facilitator (Appendix B), building enough trust so that teams feel comfortable sharing what they are experiencing is essential. Thus, it was critical to have a facilitator who members could rely on for guidance, not as an evaluator. For the teams to reflect well on what had worked well and what could have been improved in the sprint review and retrospective, they needed a good relationship between them and the facilitators.

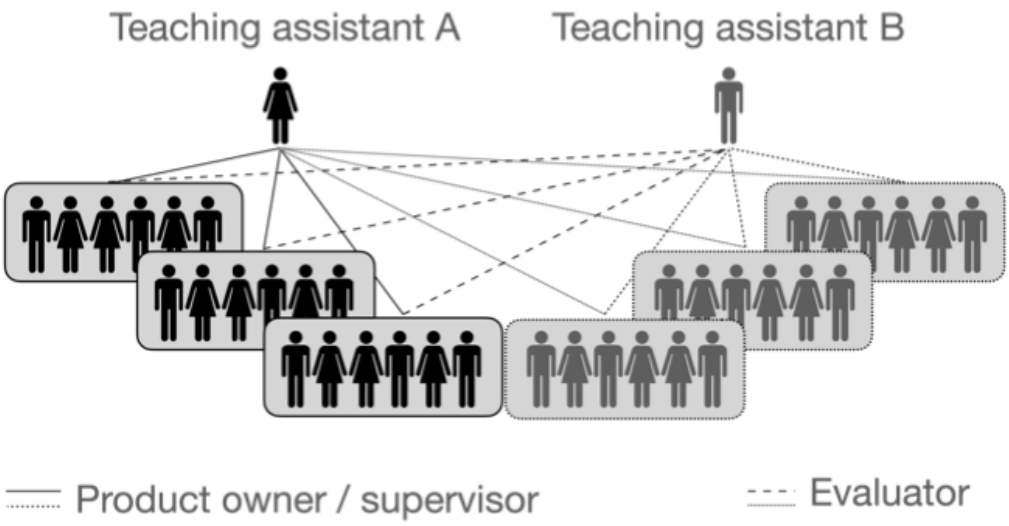


Figure 6: How roles are distributed among teaching assistants. Retrieved from (Farshchian et al., 2022)

.

5 Results

In this section the results from the case described in Section 4 will be presented. The results are derived from conducted interviews, and observations as described in Section 3.3. After analyzing the collected data, statements from both facilitators and students in the course *TDT4140 Software Engineering* were extracted. This section is structured in a way corresponding to the research question. Firstly, Section 5.1 will present results on how the use of Scrum affected development of teamwork skills. Secondly, Section 5.2 will present the findings on team facilitation provided to the students. Thirdly, identified challenges within teamwork will be presented in Section 5.3 and lastly, Section 5.4 presents the results on important skills and traits contributing to improve teamwork.

5.1 The Use of Scrum

Scrum and XP were used by students in the project. From XP, all students were required to practice pair programming with each other, preferably at least once with all team members. During their two iterations, students followed the Scrum framework, completing several activities such as sprint planning, daily Scrum and retrospective. In addition, Scrum artifacts such as product backlog, sprint backlog and "definition of done" was something the students got to experience during the project timeline. For students who lacked experience with teamwork and agile processes, some of these events and artifacts were initially perceived as time consuming and sometimes also unnecessary.

> *"At first I thought stand up meetings were a waste of time, but I have experienced that you get something in return for it. I can see how feedback from the others in the team gives more control....There are meetings almost everyday. There are a lot of meetings, but it's very good, although not all meetings are equally important, it's good to keep in touch and close dialogue if something's up." - Student 3*

Retrospective

The retrospective was reported by all informants (S1, S2, S3, S4, S5, S6) to have been the most valuable Scrum event for improvement of teamwork skills. Through observations made during the weekly meetings, feedback from the students further confirmed this. The retrospective meeting opened up an arena where team members

were able to be honest with their experiences and feelings. The analysis showed that several students (S1, S3, S4) found the retrospective to be more valuable than they initially thought. For students who lacked previous experience, the Scrum retrospective meeting was initially perceived as something very formal that they were required to complete as part of their deliveries. However, students reported that the retrospective meeting had contributed to building trust and honesty in the team. In addition it was reported that as a result of the meeting, bad habits accumulated within the team were broken which had improved teamwork.

> *"People dared to be honest with how they felt the teamwork had gone. You might think that it's just something you do because we have to, formally, but there were things that we realized that we have actually taken to heart. Several things we talked about during retrospective 1 have improved the teamwork. So I might think that this is the activity that I feel has been most effective for the teamwork." - Student 4*

The facilitators did not participate in any of their teams' retrospective. Although the facilitators were not involved in any of their teams' retrospective meetings, they were involved with evaluating the reports of other teams' retrospective meetings. Following these reports, facilitators gained an understanding of the challenges and areas of improvement faced by each team. The researchers' observations revealed that defining and clear action points for future improvements was a challenge that teams faced. By failing to properly define action points to improve the development process of the team, bad habits acquired from previous sprints were difficult to reverse. Therefore, defining clear and achievable action points was desired. Statements collected from the data analysis were made by informants after the teams' first retrospective and before their second and last retrospective meeting. Based on observations of the reports delivered of the last retrospective meeting it was evident that action points were more clear and better defined.

> *"I feel that what many teams struggle with is to clearly define points of action from the retrospective. It is actually something that is easier to follow up, and for example, many say that one must improve communication and such. Then you actually have to say how to do it. If not, then it is difficult to actually improve it. So I feel that the challenge is to specify the points well enough that they get a good enough benefit from the action points for the next iteration." - Facilitator 5*

5.2 Team facilitation

On a weekly basis, facilitators met their teams in 20-40 minute meetings. The teams gave the facilitator an update on their work, and the facilitator gave the teams feedback on their work process, and answered questions regarding their work and upcoming deliverables. The interviews revealed general satisfaction with the facilitators' efforts to get acquainted with their work. However, some marks were made on the participation of the facilitator. One of the interviewees (S2) expressed that their team had experienced difficulties with establishing good communication in the beginning of the project phase. The student seeks for more guidance given by the facilitators around communication and how to make the Scrum process work better. It was noted by a student from another team that they wished their facilitator was more supportive in encouraging their team to be social. Either suggesting the team to hang out after school hours or encouraging them to work more physically rather than digitally.

> *"If a group is generally struggling with collaboration within the team the facilitator should maybe give some guidance around it, and how to make Scrum work better and what aspects around it that one should include in the group work." - Student 2*

> *"In teams that do not have a great social environment, it might have been nice if the facilitator or the course staff to encourage something like "You should do something social" or something like "You should actually meet physically" so that all teams meet physically the first time and write that group report together. I have the impression that many groups have had an awkward atmosphere in the start of the course and often it continues beyond and affects the teamwork to a great extent." - Student 4*

Facilitators get guidelines from the course staff, but their work methods would vary to some degree. At the start of the course, all teaching assistants in *TDT4140 Software Engineering* received a handbook including overall guidelines and descriptions on how to act and behave as a facilitator (Appendix B). In addition, before each weekly meeting, facilitators were given a document with a suggested agenda which contained questions and topics that might be useful to discuss with the teams. The interviews with students revealed that many of the student teams measured their satisfaction with their facilitator by how much guidance they provided towards upcoming deliveries. Students were especially concerned about the availability of the facilitator. Responding quickly and being reachable outside of scheduled meetings was revealed as highly valued by students. Right before deliveries facilitators ex-

pressed an increased amount of inquiries made by students. Observations made by the researcher also revealed that students mainly focused on receiving information about deliverables during meetings. As soon as teams received an evaluation of their newly delivered assignment, they were eager to discuss the results with the facilitator, cutting down on facilitation time.

> *"It is important that the facilitator is available, and shows that they can be reached by e-mail outside of those weekly meetings and respond quickly." - Student 3*

> *"[TDT4140 Software Engineering] has a lot like that a bit vague. What shall I say? Assessment criteria, and it is probably on purpose, so much of what we have done has been based on feedback from the facilitator and what they have said about how to write a reflection paper or retrospective and such. After all, the facilitator has had a key role in us understanding how we are being evaluated." - Student 2*

As described in the handbook given to all teaching assistants, the role of the facilitator is defined as *"To guide the group through the project, with a focus on learning goals, deliveries and the process part of the project"*. How the facilitators chose to guide each team was to some degree open for interpretation. In a lot of cases revealed by the conducted interviews, facilitators often stepped back when decisions were being made and did not interfere as much in the staring phase pf the project. Rejection to participate more actively in the staring phase of the project can have affected the teamwork developed in the teams. Much of the guidance given by the facilitators was revealed to be focused around team dynamics and them conveying the importance of a good social environment.

> *"Sometimes the facilitator could be there to ask the right questions to get the discussion started. Not necessarily needing to involve the facilitator in absolutely everything, but there is a lot of things in the starting phase that they probably could have been guided by us a little more, and they would have stopped banging there head in the wall when trying to figure out technical things but also completely new things." - Facilitator 3*

Facilitators revealed some dissatisfaction with bearing two types of roles for each team. For each weekly meeting it was the facilitators job to distribute time between being both the product owner and facilitator. Making a clear distinction between the two roles was revealed to be a challenge for the teaching assistants.

> *"We have meetings where I am, both product owner and facilitator in the*

same meeting, so sometimes you can notice that the distinction between the two is a little weakened by it" - Facilitator 3

5.3 The Challenges of Developing Teamwork Skills

Students and facilitators were asked how they could improve, but also secure good teamwork in each team. Some teams expressed satisfaction when asked how they had experienced teamwork in their team. However, not all teams experienced equally well-functioning teamwork. This section will present challenges students and facilitators encountered during the course of the project.

5.3.1 Leadership and Structure

Lack of proper leadership was revealed as a challenge when interviewing students. The interviews revealed that the Scrum master role often went on rotation. The tasks and responsibilities of the Scrum Master were often unclear. Other than the rotating Scrum master, there few defined roles. However this can be explained by the requirement to pair program with everyone in the team, making it difficult to keep fixed role definitions. Undefined roles, affected the leadership in the teams. Lack of a clear leader resulted in a fluid structure and unstructured teamwork. It can seem like the focus on developing relationships and a comfortable atmosphere in the teams affected the level of authority. Lack of authority and unstructured work plans resulted in students not knowing what to do when showing up to work sessions and meetings being to unstructured to do anything productive.

"To have a more a defined team leader. We elected a team leader at the first meeting, but in practice there has been no leadership function at all. It's just been very fluent really. And it quickly results in the teamwork getting a little unstructured." - Student 1

"We became quite comfortable with each other and it ended up with people not always remembering to tell the team if they did not show up to a work session or a meeting and it ended up with us not always knowing what to do when we came to work sessions and meetings. There was a pretty chill atmosphere in the group." - Student 4

5.3.2 Communication

All students were grouped into interdisciplinary teams of six to eight individuals. In many cases, this was the first time students had worked in an interdisciplinary team. Interviews revealed that facilitators and students found it challenging to keep frequent communication within the team. Two main communication challenges were revealed: Poor communication regarding attendance and lack of communication when feeling stuck and in need of support. The first of these challenges was mainly a result of poor structure and lack of authority in the team as described in Section 5.3.1. The second challenge stemmed from the interdisciplinary nature of the teams. Due to differences in skills and knowledge, members of the team failed to ask for help when in need of support. This lack of communication led to a halt in the work, affecting the efficiency of the team. The cause of this communication challenge was revealed to be lack of trust between team members, but also irritation among members of the team with a higher level of competency.

> *"What has become visible over time is that the competency level within the team is different. Some do not know as much as the rest, and at the same time some of the people who know a little less are afraid to show it. So it can be a bit of a halt in the work because they do not want to ask for help, and it is a pity, but it is something I have noticed, at least in recent weeks"- Student 1*

5.3.3 Various Levels of Competence

Different levels of competence were reported to be a challenge for the interdisciplinary teams in *TDT4140 Software Engineering*. As one facilitator describes (F2), it is essential that team members remain flexible and maintain a positive mindset in order to meet the challenge. Other than a positive and flexible mindset, knowledge sharing within each team was shown to be a big contributing factor to evening out the differences in competencies. Knowledge sharing was practiced by all teams through pair programming, where students were required to rotate with each other throughout the project. Observations and interviews revealed that pair programming contributed to secure progression and increase knowledge in the team, but it also perceived to increase the level of trust between team members. All of which resulted in improved teamwork in the interdisciplinary teams.

> *"If you're a racer at programming, and you are put in a team with people that are not as good as you and you can not be flexible or positive, then*

I think you quickly get annoyed and frustrated that they are not able to perform on the same level as you or that you have to spend a lot of your own focus time helping them or training them." - Facilitator 2

"I feel that pair programming is an exercise that "gains" the team and the individual parties in programming in many ways. You get to know both the person you are sitting and pair programming with. Given that one is better than the other, you also learn a lot and from teaching. You get an extra set of eyes on the code, so you might avoid bugs. Instead of everyone sitting separately reading things, you can rather find out those things together. It also helps to build relationships and build trust in each other and also build code." - Facilitator 3

"I think pair programming has been very good for the team, because there has been quite a big difference in the competence within the team. So just sitting next to each other, has been especially important to both secure good progression, but also because you learn a lot at the same time." - Student 6

Most students stated that their team used the start of the first iteration to learn basic skills and accumulate knowledge to be able to develop the product requested by their product owner. Where there were big differences in knowledge, the teams became very vulnerable if the team member with the most knowledge and skills failed to show up. One of the facilitators (F1) reported an incident where the team member with the highest technical knowledge resigned from the course leaving the rest of the team behind. The sudden resignation was non-dramatic in the sense that it stemmed from lack of remaining personal time required by the course. However, when this happened the team struggled to move on because that team member was the one possessing the most technical skills required for the project. This incident gave the team a small delay, in which they had to restructure and adapt to the loss of knowledge.

5.4 Essential Teamwork Skills and Traits

The analysis revealed many different skills and traits that students and facilitators think are most important in well-functioning teamwork. When the data was collected students and facilitators were over halfway in the project schedule, allowing the responses to reflect their experiences with teamwork so far. Table 3 provides an overview over skills and traits students and facilitators see as most important to

develop and secure well-functioning teamwork. Students and facilitators are marked using the same identifier as in the overview of conducted interviews (Table 2). From the analysis of the qualitative data, interesting descriptions given by informants about trust and adaptability were made, which are presented in this section.

Table 3: Overview of what skills and characteristics are important to facilitators and students to secure well-functioning teamwork.

Skill	**Student**	**Facilitator**
Leadership	S1	
Kindness	S1	
Trust	S2, S5	
Engagement	S3	F1, F5
Adaptability	S4	
Transparency	S1, S2, S6	F3
Flexibility		F2
Honesty	S2, S6	F3
Cooperative		F5
Positivity	S4	F1, F2
Structure	S4	
Motivation		F5
Self-driven		F3
Respect	S6	F4
Give positive feedback	S4, S5	

Trust

Trust was described by informants (S2, S5) to be an important quality to secure good teamwork. As presented in Section 5.3.2, lack of trust was one of the contributing factors when team members failed to ask for help. To increase the level of trust in

the team, giving positive feedback and backing each other was reported to help.

One of the students made an interesting reflection regarding the need to make decisions individually and to have sufficient trust in one's team to deal with the consequences if the decision turns out to be wrong:

> *"Being able to ask for help, and being good at backing each other. There are many choices that must be made by individuals and that affect the product to a fairly large extent, and this means that we can trust that it is possible to make the wrong choice and that we are good at telling each other about it quickly. You also have to give positive feedback for it, otherwise it will be very harsh if you only give feedback when something goes wrong."- Student 5*

For facilitators it was important to establish trust with their team. To guide the team in the best way possible and enable them to open up and talk about their challenges, facilitators were dependant on establishing trust with the team. For one of the facilitators (F3) this was a priority from the start. Asking the right questions and forcing them to think about parts of their work was reported to be important to secure teamwork.

Adaptability

Today's economic system has resulted in businesses seeking greater versatility and adaptability, which has, in turn, affected the competencies they require (Villagrasa and Conchado, 2018). Being able to adapt quickly and being a flexible team member was argued as valuable qualities to possess before graduating from university. A statement made by an informant (S4) on which skills and traits are most important in fostering good teamwork, revealed adaptability to be the most important. To be adaptable was described as being able to concentrate on specific tasks, keeping up with deadlines, creating relationships within the team and on top of that thinking critically and not being afraid of giving feedback. This adaptability was considered to be incredibly valuable for the team.

> *"That you are adaptable, both that you can focus on a task, and also can focus on the structure of the work, and especially in this type of work where we work in sprints and things like that and have a lot of deadlines, but also that you are a social person, you want to create relationships with the others on the team, and not just be concerned with the task, and that you dare to ask some critical questions and dare to give praise to the others." - Student 4*

6 Discussions

In this section, the results from Section 5 will be discussed. While discussing the results, the focus will be around the research question: *How can facilitators Contribute to Improve Teamwork Skills through Facilitating Project-Based Learning in Software Engineering Education?* The main goal of this section is to discuss and investigate how facilitators can contribute to improving students' teamwork skills. We will begin by discussing how the use of Scrum affected the improvement of teamwork skills. Following this, we will discuss the role of the facilitator and their influence on teamwork followed by a discussion about challenges identified with teamwork and how facilitators can counteract them. Finally, some limitations identified during the analysis will be explained.

6.1 The Use of Scrum

6.1.1 Retrospective

The retrospective was organized and conducted by the teams without the facilitator present. Scrum retrospectives and other Scrum events serve to promote transparency as well as allow for inspection and adaptation of Scrum (J.Sutherland, 2020). Sutherland describes continuous improvement as a key component of the retrospective in the Scrum Guide (J.Sutherland, 2020). Through discussing areas of improvement and defining action points for the next sprint, teams can improve efficiency, quality, and teamwork. As revealed from the results, student informants in this case study rated the retrospective as the most valuable event for improving teamwork. The findings from the conducted interviews were supported by observations and feedback from students during the weekly facilitation meetings. Some students were initially sceptical of the Scrum event since they had no previous experience with retrospectives. In spite of this, students expressed their satisfaction after experiencing their first retrospective and the effect it had on the next iteration.

In a study by Matthies et al. on how to counteract problems arising during retrospective meetings, facilitators and coaches identified common problems faced by their teams (Matthies et al., 2019). Further on, they were provided with activities that would counteract the identified issues their teams faced. One of the challenges reported in Section 5 was the specification of action points. Observations made by the researcher also revealed it to be difficult for students to make clear and defined

action points. As suggested by Matthies et al., facilitators and coaches can provide support to help prevent teams from meeting common challenges. From this point of view, it is possible to argue that participation in retrospective meetings for facilitators in *TDT4140 Software Engineering* could potentially reduce the challenges students face in these meetings. By improving the effectiveness of the retrospective meeting one can also argue that facilitators contribute to improving the teams' teamwork skills.

6.2 Team Facilitation

From the analysis and results (Section 5) I discovered that the facilitator could be more involved in establishing proper collaboration in the team. Collaboration is established as an important teamwork skill, and failure to secure collaboration between team members can greatly affect the teamwork (Largent, 2016). Collaboration is also one of four traits included in the 4C's of 21st Century Learning, which are essential to understand the mental processes required to self-develop in a modern working environment (Chiruguru, 2020). For students in the *TDT4140 Software Engineering*, developing collaboration skills is especially important as they have little previous experience with teamwork, and their first experiences will have a greater impact on their future development as team members in the 21st century.

Oakley et al. argue that putting students in teams and letting them manage their work themselves is not sufficient to achieve collaborative learning (Oakley et al., 2004). An instructor must ensure that the team is able to develop the necessary skills to work effectively and secure team learning. He suggest that making a common declaration of expectations will contribute to establish a realistic set of expectations that will unite the team. In *TDT4140 Software Engineering*, students were required to make a declaration of expectations with their team at the start of the course. However, this was done before the first meeting with their facilitator. Oakley et al. also suggests that to prevent challenges in establishing good collaboration upon team formation, students may benefit from being informed on typical mistakes commonly made by new teams, and how to avoid them before making the declaration of expectations (Oakley et al., 2004). Getting the facilitator involved earlier while the declaration of expectations is created can therefore be beneficial for the team. Also, as suggested by Oakley et al., the facilitators should inform teams on common mistakes new team often make, to help them prevent common pitfalls.

It can be argued that the facilitator should become more involved as students in the course *TDT4140 Software Engineering* lack previous experience with teamwork

and interdisciplinary teams. Agile coaches are often used in organizations when adapting to agile methods and development of agile teams (Daljajev et al., 2020). The need for team coaching might be increased in some teams then other, based on team members prior experiences. The results showed that only two students (S2 and S4) from distinct teams sought for increased involvement from the facilitator. This may be a sign that the level of facilitator involvement varies from team to team depending on the team members' previous experience with team coordination, or it may be because of different facilitation provided by the facilitators. Either way, we can argue that lack of involvement and guidance from the facilitator have affected the development of important teamwork skills in these teams. From the findings this is also indicated by one of the facilitators (F3): *"....there is a lot of things in the starting phase that they probably could have been guided by us a little more, and they would have stopped banging there head in the wall...."*. Current literature support this statement and suggests that to assist teams with the needed support in development of teamwork skills, team facilitators can have specific conversations with the teams and make sure that they are aware of common pitfalls in teamwork (Widdowson et al., 2020).

Through both interviews and observations, it became apparent that students placed a great deal of emphasis on receiving information regarding deliverables and discussing their evaluations during their weekly meetings with their facilitator. Thus, reducing team and process facilitation time. As a result, the distinction between the role of facilitator and evaluator became more unclear, possibly affecting the facilitation provided to the teams. In order to provide proper facilitation that will enhance teamwork, facilitators must make sure students understand the difference between facilitators and evaluators. Time spent on discussing evaluations only takes away time from coaching their teams and improving their teamwork.

6.3 The Challenges of Developing Teamwork Skills

The findings from Section 5 presented challenges with the development of teamwork skills regarding defined leadership, communication and various levels of competencies. This section will discuss these challenges and present ways in which facilitators can counteract the identified issues.

6.3.1 Leadership and Structure

The results revealed that undefined roles and unclear leadership affected the structure of the work in teams. The student teams in *TDT4140 Software Engineering* are required to self-organize and manage their work like any other agile autonomous team. In an article on challenges in agile teams, Stray et al. identifies lack of coaching and organizational support as one of the challenges autonomous teams face (Stray et al., 2018). In an autonomous team the leadership is distributed among a group of individuals instead of being assigned to a single individual. For a new team with little prior experience Stray et al. also suggests that leaders must guide the team in the right direction before they can take on a more coaching role (Stray et al., 2018). A similar argument is made by Vogler, who argues that if students are to develop teamwork skills, they need scaffolding from the facilitator (Vogler, 2018). Results showed that lack of knowledge about organizing and self-managing led to undefined leadership and roles within teams that reported unstructured work environments. As suggested by Stray et al. it can be beneficial for the facilitator to inform the teams of the known challenges of leadership and team organization to guide them in the right direction (Stray et al., 2018). For teams in *TDT4140 Software Engineering* this could be particularly helpful in the start-up phase of the project. By helping the teams establish a good structure and leadership right from the start, they will have better support and a better chance of improving their teamwork skills.

6.3.2 Communication

By communicating early about a given problem, teams are more likely to resolve them together and quicker (Largent, 2016). The findings revealed that a challenge some teams faced was lack of communication when in need of help and support. The interdisciplinary nature of the teams contributed to increase differences in competencies within the teams. For one of the teams, lack of trust and irritation among team members with higher technical skills resulted in poor communication within the team. In fear of being judged as less competent and knowledgeable, some students were revealed to not dare to ask for help, which resulted in a halt in communication. As Largent describes in his article on team development, lack of communication between team members may result in team members distrusting them as they have broken their promises in the past, simply because they did not realize anything was wrong until it was too late to take action (Largent, 2016). Being able to ask for help when in need of it can therefore be argued as vital to be able to retain good

teamwork.

Trust, transparency and honesty were among the qualities informants of this study reported as the most important when maintaining good teamwork (Table 3). Developing these features within a new team where nobody is acquainted can be time consuming. Therefore, the facilitators should inform the team of the importance of establishing an environment where everybody can speak freely and express themselves. In order to develop trust and honesty within a team, one of the facilitators emphasizes the importance of asking students the right questions and forcing them to think about their teamwork: *"Sometimes the facilitator could be there to ask the right questions to get the discussion started."* (F3). Considering the statement, we can argue that increased facilitator involvement is beneficial in improving communication within a team.

6.3.3 Various levels of Competence

To achieve successful teamwork in interdisciplinary teams, Vogler argues that utilizing your technical and soft skills is not enough (Vogler, 2018). Students require scaffolding from facilitators to be able to develop the necessary teamwork skills (Vogler, 2018). One of the challenges reported by both students and facilitators was the various levels of competencies within each team. Working in interdisciplinary teams introduces many benefits as presented in Section 2.4.2, but also some challenges. As discussed in Section 6.3.2, the interdisciplinary nature of the teams prevented some students from asking for help when in need of support. Increased differences in competencies within the teams had contributed to some members fearing they would be judged as less competent and knowledgeable if they asked too many questions. Another challenge that arose as a result of the interdisciplinary environment, was differences in knowledge and skills. In an interdisciplinary team everyone has different skills and capabilities. The ability to utilize these skills in the best way possible will provide students with valuable experiences and the opportunity to develop important teamwork skills (Vogler, 2018). One of the facilitators (F3) emphasized the need for students to be both positive and flexible when facing differences in competencies. If you are more equipped with technical skills you should maintain a positive attitude towards your team members with less technical skills. The other way around team members feeling that they lack necessary technical skills should be flexible and find other ways to contribute. Making teams aware of this can improve teamwork skills, and is therefore something facilitators should inform students of.

To reduce differences in technical skills and increase knowledge sharing, Kniberg

argues that pair programming is a good practice (Kniberg, 2015). Students were required to pair program with each other during the project time. In agreement with Kniberg students reported satisfaction with pair programming through the course duration. In addition to evening out technical differences, pair programming was also observed to increase the level of trust and building relationships within the team. Based on the reported advantages of pair programming, facilitators should continue to promote this practice as it is effective in achieving knowledge sharing and better teamwork.

6.4 Limitations

This study is not without limitations. The results and discussions are based on results from the interviews and observations conducted as part of this case study. Considering the limited number of interviews, it is possible that this may have affected the analysis and the results. It is also possible that the participants have a bias and have more interest in teamwork than others. It is possible that the conclusions we have drawn from the interviewees in this case do not accurately represent what other informants would say in the same circumstances.

Observations were conducted by the researcher of this thesis, who also worked as a teaching assistant in the investigated course. The observations included following three student teams through the semester and evaluating an additional three teams. The limited number of teams observed and evaluated can have affected the validity of the results in this study. It is possible that the conclusions made from the observations would not have been the same if the observations were conducted on other student teams.

An analysis of the data revealed several success factors for team development with no relevance to the research question, which puts a special focus on how facilitators can contribute to team development. These factors included improving teamwork in general, in which the facilitator will not be able to contribute. The factors were therefore omitted from the results and irrelevant for discussion.

In this thesis, the literature review revealed that previous studies on facilitation in software engineering education are limited. The lack of published literature has therefore limited our ability to discuss and investigate experiences and empirical data from other studies on how facilitation can improve teamwork skills through PBL.

7 Conclusion

This study investigated how facilitators in software engineering education can contribute to improve students teamwork skills. The results are based on a case, where both facilitators and student have been interviewed and observed during the course of a project-based learning course using the agile methods Scrum and XP. Findings indicated that increased team facilitation is beneficial in development of teamwork skills for undergraduate students. By participating more actively during the beginning of the project, in order for students to establish proper communication and team structure, facilitators can contribute to improving students' teamwork skills. It is also suggested that by participating in the students retrospective meeting, facilitators can reduce challenges and increase the effectiveness of the meeting. Students' lack of experience with teamwork in interdisciplinary teams also suggests that facilitators should become more involved. By having specific conversations with their teams, facilitators can help teams become more aware of common challenges they might face.

The investigation revealed challenges with leadership, communication and different levels of competence. By informing their teams of known leadership and team organization challenges, facilitators can help prevent these challenges from occurring. Having enough trust between the facilitator and within the team is also revealed as important in the process of improving communication and teamwork skills. Without open and transparent conversations with the teams, the facilitator will not be able to provide proper coaching. Pair-programming was reported to build trust and relationships with in teams, in addition to being a great activity of knowledge sharing. Continuous promotion of this activity was concluded to have a positive effect on leveling out differences in competence and improving teamwork.

This study has revealed what facilitators can do to improve students' teamwork skills. The use of project-based learning and interdisciplinary teams has been shown to encourage the development of teamwork skills. However, as this study shows, some scaffolding from facilitators is needed for these teamwork skills to be improved. By becoming more involved and aware of the challenges teams face during the development process, the facilitator can contribute to the development of better teamwork skills such as communication, collaboration, and leadership.

7.1 Contributions

This master thesis identifies what facilitators can do to improve students' teamwork skills in software engineering education. Previous literature on facilitation in software engineering education is limited, and therefore this case study will contribute with empirical evidence on that topic. The thesis discusses how facilitators can improve development of teamwork skills in addition to challenges with teamwork in a project-based learning course. This thesis can hopefully be an inspirational source for course staff or practitioners interested in improving teamwork.

7.2 Future Work

In light of the findings and discussion of this thesis, further research on facilitation and development of teamwork skills in software engineering education is recommended, since this type of research is lacking in the existing literature. In addition, agile leadership and coaching are also interesting topics for further research, as they may explain how facilitators and student teams can succeed with teamwork. Research on agile leadership can provide an in-depth understanding of how student teams can be managed and structured based on the principles of effective leadership. Furthermore, research on Agile coaches' involvement in software engineering education can increase the understanding of how students with only traditional classroom experiences can adapt to project- and team- based learning.

It would be interesting to conduct a longitudinal ethnography of similar cases to the one in this study. This could provide a deeper and more comprehensive understanding of how facilitators can improve the development of teamwork skills and how to face challenges during the process. This type of study should include undergraduate software engineering students who have little prior experience with teamwork and follow them as they gain skills over time at university. It is likely that this type of study will provide a more in-depth understanding of the challenges facilitators and students encounter during the process of improving teamwork skills. Additionally, it would be interesting to look at the challenges these students face in their senior years as they have gained more experience with teamwork.

Bibliography

Almestad, J. & Olssen, H. (2021). *How to improve development of teamwork skills in software engineering education through project-based learning with agile methodologies* (tech. rep.). Norwegian University of Science and Technology (NTNU).

Vogler, e. (2018). The hard work of soft skills: Augmenting the project-based learning experience with interdisciplinary teamwork. *INSTRUCTIONAL SCIENCE*, *46*, 457–488.

Vicente, e. a. (Ed.). (2020). *Methodology for training engineers teamwork skills*. IEEE.

Tkalich, A. (2020). *Agile coaching in norway and the usa* (tech. rep.). Norwegian University of Science and Technology (NTNU).

Bäcklander, G. (2019). Doing complexity leadership theory: How agile coaches at spotify practise enabling leadership. *Creativity and Innovation Management*, *28*, 42–60. https://doi.org/10.1111/caim.12303

Bancino, R. & Zevalkink, C. (2007). Soft skills: The new curriculum for hard-core technical professionals. *The skills for the job*, *82*, 20–22.

Gutica, M. (Ed.). (2018). *Improving students' engagement with large-team software development projects*. ACM - Digital Library.

Mishra, A. & Mishra, D. (2012). Industry oriented advanced software engineering education curriculum. *Odgojne Znanosti*, *14*, 595–624.

Su, H., Jodis, S. & Zhang, H. (2007). Providing an integrated software development environment for undergraduate software engineering courses. *J. Comput. Sci. Coll.*, *23*(2), 143–149.

Loftus, C., Thomas, L. & Zander, C. (2011). Can graduating students design: Revisited. https://doi.org/10.1145/1953163.1953199

Mishra, A. & Yazici, A. (2011). An assessment of the software engineering curriculum in turkish universities. *Croatian Journal of Education*, *13*, 188–219.

Cico, O., Jaccheri, L., Nguyen-Duc, A. & Zhang, H. (2021). Exploring the intersection between software industry and software engineering education - a systematic mapping of software engineering trends. *JOURNAL OF SYSTEMS AND SOFTWARE*, *172*. https://doi.org/10.1016/j.jss.2020.110736

Ghezzi, C. & Mandrioli, D. (2006). The challenges of software engineering education, 115–127. https://doi.org/10.1007/11949374_8

Warr, M. & West, R. E. (2020). Bridging academic disciplines with interdisciplinary project-based learning: Challenges and opportunities. *INTERDISCIPLIN-*

ARY JOURNAL OF PROBLEM-BASED LEARNING, *14*. https://doi.org/10.14434/ijpbl.v14i1.28590

Babb, J., Hoda, R. & Nørbjerg, J. (2013a). Toward learning teams. *Software, IEEE*, *30*, 95–98.

Babb, J., Hoda, R. & Nørbjerg, J. (2013b). Barriers to learning in agile software development projects, 1–15. https://doi.org/10.1007/978-3-642-38314-4_1

Coleman, G. & O'Connor, R. (2008). Investigating software process in practice: A grounded theory perspective. *Journal of Systems and Software*, *81*, 772–784. https://doi.org/10.1016/j.jss.2007.07.027

Babb, J., Hoda, R. & Nørbjerg, J. (2014). Embedding reflection and learning into agile software development. *Software, IEEE*, *31*, 51–57. https://doi.org/10.1109/MS.2014.54

J.Sutherland, K. S.

bibinitperiod. (2020). The 2020 scrum guide [https://scrumguides.org/scrum-guide.html].

Uskov, V., Krishnaiah, D. B., Kondamudi, R. & Singh, U. (2016). Innovative agile project management curriculum for engineering education [Cited By :3]. *Ieee global engineering education conference, educon* (pp. 463–468). www.scopus.com

Wallace, C., Mohan, S., Troy, D. & Hoffman, M. E. (2012). Scrum across the cs/se curricula: A retrospective, 5–6. https://doi.org/10.1145/2157136.2157142

Matthies, C., Dobrigkeit, F. & Ernst, A. (2019). Counteracting agile retrospective problems with retrospective activities. https://doi.org/10.1007/978-3-030-28005-5_41

Kniberg, H. (2015). *Scrum and xp from the trenches*. C4Media.

Blom, M. (2010). Is scrum and xp suitable for cse development? In P. Sloot, G. Albada & J. Dongarra (Eds.), *Iccs 2010 - international conference on computational science, proceedings*.

Patil, S. P. & Neve, J. R. (2018). Productivity improvement of software development process through scrumban: A practitioner's approach. *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, 314–318. https://doi.org/10.1109/ICACCT.2018.8529405

Baser, D., Ozden, M. & Karaaslan, H. (2017). Collaborative project-based learning: An integrative science and technological education project. *Research in Science Technological Education*, *35*, 1–18. https://doi.org/10.1080/02635143.2016.1274723

Holvikivi, J. e. a. (Ed.). (2016). *Introducing collaborative practices to undergraduate studies*. SAITE.

Vicente, e. a. (2018). Collaborative approach in software engineering education: An interdisciplinary case. *Journal of Information Technology Education: Innovations in Practice*, *17*, 127–152.

Fisher, K. & Newton, C. (2014). Transforming the twenty-first-century campus to enhance the net-generation student learning experience: Using evidence-based design to determine what works and why in virtual/physical teaching spaces. *Higher Education Research & Development*, *33*, 903–920.

Støren, L. e. a. (2019). *Utdanning for arbeidslivet : Arbeidsgivers forventninger til og erfaringer med nyutdannede fra universiteter, høgskoler og fagskoler* (tech. rep.). The Nordic Institute for Studies in Innovation, Research and Education (NIFU).

Crawford, E. & Lepine, J. (2013). A configural theory of team processes: Accounting for the structure of taskwork and teamwork. *Academy o/ Management Review*, *38*, 32–48.

Xia, Y., Cutler, S. & McFadden, D. (2020). Collaborative project-based learning approach to the enculturation of senior engineering students into the professional engineering practice of teamwork [https://peer.asee.org/34299]. *2020 ASEE Virtual Annual Conference Content Access*, (10.18260/1-2–34299).

Largent, D. L. (2016). Measuring and understanding team development by capturing self-assessed enthusiasm and skill levels. *ACM TRANSACTIONS ON COMPUTING EDUCATION*, *16*.

Chong, J. & Hurlbutt, T. (2007). The social dynamics of pair programming. *29th International Conference on Software Engineering (ICSE'07)*, 354–363. https://doi.org/10.1109/ICSE.2007.87

Cerato, A., Elton, D. & Shannon, D. (2012). Building student teamwork with the student geo-challenge. *JOURNAL OF PROFESSIONAL ISSUES IN ENGINEERING EDUCATION AND PRACTICE*, *138*(1), 14–20. https://doi.org/10.1061/(ASCE)EI.1943-5541.0000082

Thompson, A. (2017). Fostering development of teamwork skills in an introductory engineering course [IEEE Frontiers in Education Conference (FIE), Indianapolis, IN, OCT 18-21, 2017]. *2017 IEEE FRONTIERS IN EDUCATION CONFERENCE (FIE)*.

Yusof, K. M., Hassan, S. A. H. S., Jamaludin, M. Z. & Harun, N. F. (2012). Cooperative problem-based learning (cpbl): Framework for integrating cooperative learning and problem-based learning. In Y. Yusof & F. Phang (Eds.), *International conference on teaching and learning in higher education in conjunction with regional conference on engineering education and research in higher education* (pp. 223–232). https://doi.org/10.1016/j.sbspro.2012.09.649

Terenzini, P. T., Cabrera, A. F., Colbeck, C. L., Parente, J. M. & Bjorklund, S. A. (2001). Collaborative learning vs. lecture/discussion: Students' reported learning gains. *Journal of engineering education (Washington, D.C.)*, *90*(1), 123–130.

Johnson, D., Johnson, T. & Stanne, M. (2000). Cooperative learning methods: A meta-analysis.

Oakley, B., Brent, R., Felder, R. & Elhajj, I. (2004). Turning student groups into effective teams. *Journal of Student Centered Learning*, *2*.

Chiruguru, S. (2020). The essential skills of 21st century classroom (4cs). https://doi.org/10.13140/RG.2.2.36190.59201

Daljajev, K., Scott, E., Milani, F. & Pfahl, D. (2020). A study of the agile coach's role, 37–52. https://doi.org/978-3-030-64148-1

Rodríguez, G., Soria, Á. & Campo, M. (2016). Measuring the impact of agile coaching on students' performance. *IEEE Transactions on Education*, *59*(3), 202–209. https://doi.org/10.1109/TE.2015.2506624

Stray, V., Moe, N. B. & Hoda, R. (2018). Autonomous agile teams: Challenges and future directions for research. *Proceedings of the 19th International Conference on Agile Software Development: Companion*. https://doi.org/10.1145/3234152.3234182

Hoda, R., Noble, J. & Marshall, S. (2013). Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering*, *39*(3), 422–444. https://doi.org/10.1109/TSE.2012.30

Hoda, R., Noble, J. & Marshall, S. (2011). Supporting self-organizing agile teams - what's senior management got to do with it? *Lecture Notes in Business Information Processing*, *77*, 73–87. https://doi.org/10.1007/978-3-642-20677-1_6

Hackman, J. & Wageman, R. (2005). A theory of team coaching. *Academy of Management Review*, *30*. https://doi.org/10.5465/AMR.2005.16387885

Salas, E., Shuffler, M., Thayer, A., Bedwell, W. & Lazzara, E. (2014). Understanding and improving teamwork in organizations: A scientifically based practical guide. *Human Resource Management*, *54*. https://doi.org/10.1002/hrm.21628

Hawkins, P. (2017). *Leadership team coaching*. Kogan Page.

Widdowson, L., Rochester, L., Barbour, P. & Hullinger, A. (2020). Bridging the team coaching competency gap: A review of the literature. *International Journal of Evidence Based Coaching and Mentoring*, *18*, 35–50. https://doi.org/10.24384/z9zb-hj74

Oates, B. (2006). *Researching information systems and computing: Overview of the research process*. Sage Publications.

Johannesson P., P. E. (2014). *Research strategies and methods. In: An introduction to design science*. Springer, Cham.

Yin, R. K. (2009). *Case study research: Design and methods*. Sage.

Kaplan, B. & Maxwell, J. (2005). Qualitative research methods for evaluating computer information systems. https://doi.org/10.1007/0-387-30329-4_2

Myers, M. D. & Newman, M. (2007). The qualitative interview in is research: Examining the craft. *Information and Organization*, *17*(1), 2–26. https://doi.org/https://doi.org/10.1016/j.infoandorg.2006.11.001

Farshchian, B., Xu, C., Smith, P., Johre, T., Dingsøyr, T., Næss, M., Grimsmo, A. & Landgaard, S. (2022). Tdt4140 – håndbok for læringsassistenter.

Villagrasa, J. & Conchado, A. (2018). Competence-based education through the application of lean canvas methodology: An empirical approach [11th Annual International Conference of Education, Research and Innovation (ICERI), Seville, SPAIN, NOV 12-14, 2018]. In L. Chova, A. Martinez & I. Torres (Eds.), *11th international conference of education, research and innovation (iceri2018)* (pp. 2808–2816).

Xu, C. (2022). Product description: Groupup.

Johre, T. & Dingsøyr, T. (2022). Veiledningsmøte 3.

A Interview Guides

For this thesis the interview guides were created in Norwegian as all participants' mother language is Norwegian. Two interview guides were created, one for interviewing facilitators and the other for students.

The first round of interviews was conducted with facilitators from the course *TDT4140 Software Development* at NTNU. The teaching assistants, also bear the role as a facilitator for three teams each in the course. During the first round of interviews, it was important to identify the work of the facilitators, as well as their thoughts on teamwork, interdisciplinary teams and agile development. The second round of interviews was conducted with students from the same course and focused on the involvement of the facilitators in their teamwork, the teamwork they experienced in their interdisciplinary team and their experience with Scrum.

The interview guide is sorted in certain categories in relation to the research question: Role, Teamwork, Interdisciplinary and Agile Methods. Within each category there are a number of general questions, not intended to expect any certain answer.

Interview Guide: Facilitators

Introduction

- Welcome & thank you for participating
- What is your team number?

Role

- In what way do you as a product owner participate in the development of the product?
- As a facilitator, how to you participate and what do you contribute with to your teams?

Teamwork

- What can a facilitator do to ensure good teamwork within the teams?
- What can you do to improve teamwork in teams?

- In the weekly meetings you have with your teams, do you get a good overview of the status of their work?
- Have any conflicts arisen in any of the teams you facilitate?
  - Why have conflicts arisen?
  - Why do you think there have been no conflicts?
- What qualities and skills do you think are important for development of good teamwork?

Interdisciplinary

- How do you think the interdisciplinary environment affects the teamwork in your teams?
- What do you have to pay attention to in an interdisciplinary team to ensure good communication and teamwork?
- Have you been enrolled in either *Experts in Teamwork (EiT)* or *TDT4290 Customer Driven Project*?

Agile methods

- How do you think the methodology the teams uses contributes to improve the teamwork in the teams?
  - Do your teams use one specific methodology or a combination of several?
- What effect has the retrospective meeting had on the teams?
  - Have they solved problems

Final comments

- Do you have any last remarks or thoughts on how you as a facilitator can contribute to increase development of teamwork skills?
- Thank you for participating

Interview Guide: Students

Introduction

- Welcome & thank you for participating
- What is your team number?

Facilitation

- How does your facilitator contribute to the development of the product and your work?
- What should a facilitator do to help you as much as possible with the teamwork?

Teamwork

- How is the teamwork in your team?
  - Have you had any conflicts in your team?
    - how have you resolved them?
    - Why do you think there have been no conflicts?
  - Is there a low threshold for asking for help?
  - Do you have good communication between each other?
- How can you improve the teamwork in your team?
- Is the product you develop affected by the teamwork in your team?
- Which qualities do you think are most important for ensuring and improving the teamwork?
- Have you learned anything new about how to work in a team through *TDT4140 Software Engineering*?

Interdisciplinary

- How does the interdisciplinarity of the team affect your teamwork?
- What do you have to pay attention to in an interdisciplinary team to ensure good communication and teamwork?

Agile methods

- What agile methods do you use in your project work?
- How has scrum (or another methodology) contributed to improve the teamwork in your team?
- How have you experienced pair programming?
- What roles and responsibilities do you have in the team?
- Of all the scrum events you carry out, which one do you think contributes the most to improving the teamwork in the team?

Final comments

- Do you have any last remarks or thoughts on how the facilitator can contribute to increase development of teamwork skills in your team?
- Thank you for participating

B Role as the facilitator

As a facilitator, you are a process facilitator for the team. You are the facilitator for the same three teams for which you are the product owner. The goal is to guide the team through the project, with a focus on learning goals, deliveries and the process part of the project. By being curious and asking exploratory questions, you give the team the opportunity to think for themselves and reflect on the way they work based on the learning objectives in the course. You should not be a technical facilitator for the teams, but can answer and give recommendations on technical questions if you can. From experience, many teams will have a relatively high threshold for asking you questions and asking for help. It is often difficult to "get close" to the students' process and team dynamics. Remember to be proactive and "on" to be a good mentor who helps the teams.

It can be good to have a clarification of expectations about what the facilitator role entails:

Example

I am the facilitator of this team. My role is not to be an oracle that can give you the answer to everything you may be wondering, but to help the team become aware of their own work process. I will do this by asking questions about different aspects of your work. It is important that you are honest about how you work when we have meetings, so that I can help you in the best possible way. I'm not here to evaluate you, but to support you. We are on the same team! I will focus on learning objectives in the subject, but if progress is hindered by technical challenges, I will try to answer such questions as well. We will have weekly meetings, where both you and I have the opportunity to ask questions.

You should meet prepared for all meetings, and have an overview of what the team is working on and what possible challenges the team has. Feel free to take a look at the team's repository before the meetings to see if the group has started, or shows signs of a very skewed division of labor. Examples of questions you can ask the group:

- What challenges have you faced since the last meeting?
- How are you doing regarding the release plan?
- How do you think the total workload is?
    - If you experience a heavy workload, how do you plan to deal with it?

- How did you distribute the workload among team members?
- Have you encountered any collaboration issues?
  - How has it affected your work?
  - How did you handle these?

These are just an examples of questions you can ask. It is challenging to build enough trust that the teams are honest with themselves and you as the facilitator with the work process. It can help to often clarify that you are not here to evaluate, but to support. This is necessary for the group to reflect well on what has gone well, and what could have been better in the sprint review and retrospective.

Retrieved from (Farshchian et al., 2022), page 6-7

C Product Description: GroupUp

Product Vision

A platform intended to facilitate new social acquaintances.

Description

The first year at university can be hectic, many meet new friends for life and others find their passion. After two years of social limitations, there are several groups of friends who feel the need to do something other than the fixed patterns of activities with the same people.

I want a platform where groups of friends can meet to cultivate common interests and get to know each other. Users should be able to create groups, where they can add their friends. Thus, a group consists of one or more personal profiles. A group must be able to add their interests, have a description, age range, photo and date of the desired activity. Furthermore, they should be able to see all other groups, and filter them by interests, location, age, group size and date before the desired meeting. The groups must match with each other for a meeting to take place, and they must be able to communicate with each other. As meetings are potentially agreed, all groups should see a list of matches with other groups.

To ensure pleasant experiences on the platform, I want groups to be able to give a review to each other after the meeting date. Furthermore, to ensure serious users. Users should have to log in with the password of a personal user to participate in the forum. It must be possible for users to report other users who violate the guidelines. If the cite administrator detects a breach or fraud, it should be possible for the administrator to remove inappropriate groups and delete users.

Whether you decide to develop a mobile app or website is up to you. The overall design of the product is important.

Retrieved from (Xu, 2022)

D Weekly facilitator meeting: Agenda

This document provides a brief description of what was done at the previous guidance meeting, what the teams have done since the last time and the agenda for today's meeting. Finally, there are suggestions for questions and topics that can be addressed to support the team's development process.

Last Guidance Meeting

- Teams received oral feedback on L3 Preliminary Study.
- Groups received oral feedback on L3 Preliminary Study. Most groups were still in the start-up phase and received general follow-up on their product and from their facilitator.

Since Last Time

- Feedback on L3 Preliminary study has been uploaded to BlackBoard.

Today's Guidance Meeting

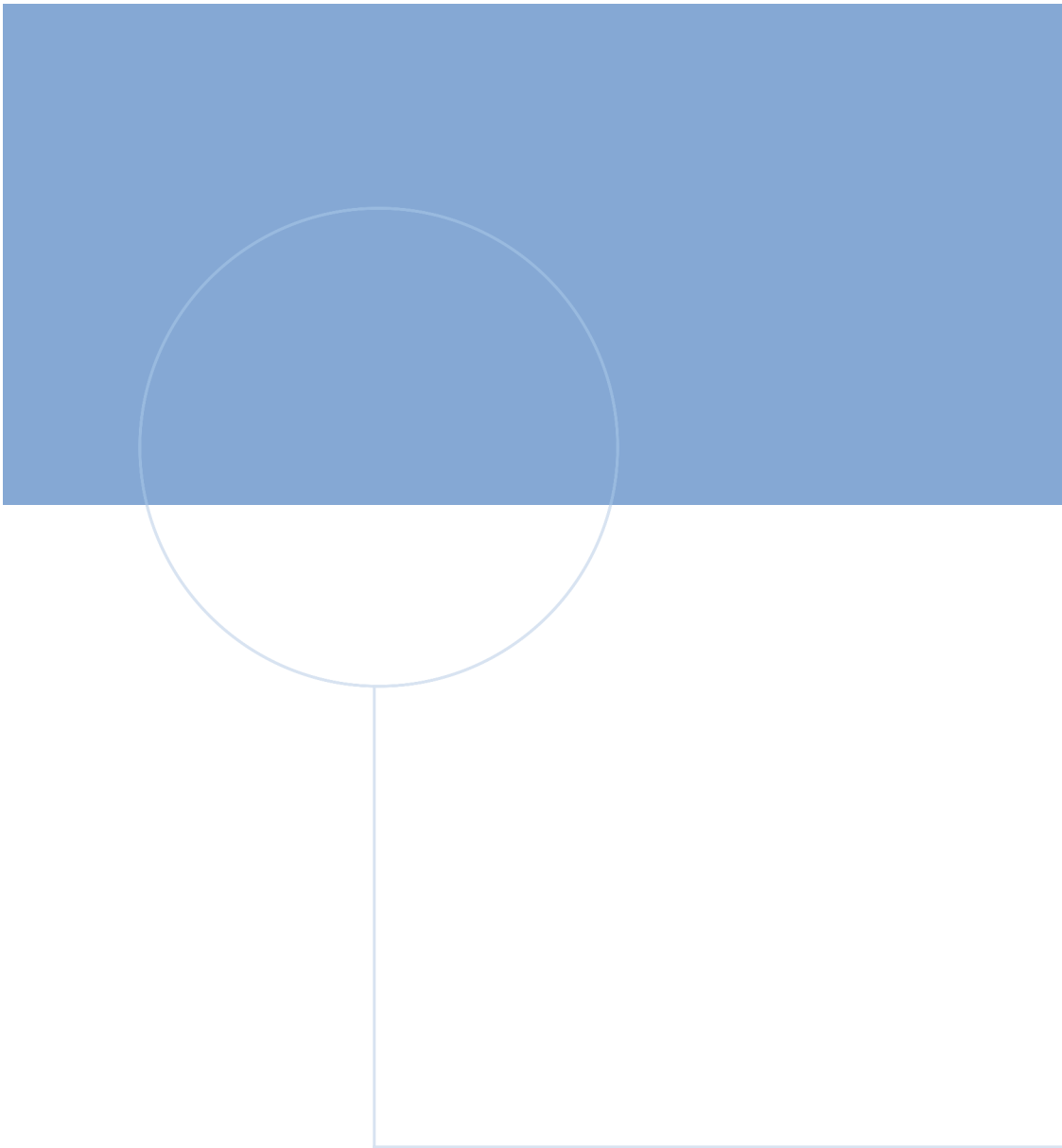
Facilitator:

- Communicate that the assignment task, rubric and time for L4 will be uploaded on Sunday. Assignment task and rubric for L5 will be uploaded at the same time. If there are any questions regarding these, we will discuss it during the next guidance meeting on Tuesday.
- Today is the last meeting before L4 and L5.
- How has the team followed up L3 Preliminary Study?
- How has the team followed up the group contract?
- What challenges has the team faced so far?
- Ask about which Scrum and XP processes the group uses. Which works well, which doesn't work so well, what adjustments has the group made?

Product Owner:

- What is the status of the product? Can you show some features?
- *Depending on what the team shows*: How does this fulfill the customer's wishes?
- What will the team demonstrate at the review in two weeks?
- Is there anything that is unclear about the product?

Retrieved from (Johre and Dingsøyr, 2022)



NTNU

Kunnskap for en bedre verden