Cornelia Reme-Ness
Rebecca Sandstø

Successful Implementation of Agile Practices in Norwegian Projects: A Case Study

Master's thesis in Mechanical Engineering
Supervisor: Bassam Hussein and Torbjørn Kågen
June 2021

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Cornelia Reme-Ness
Rebecca Sandstø

Successful Implementation of Agile Practices in Norwegian Projects: A Case Study

Master's thesis in Mechanical Engineering
Supervisor: Bassam Hussein and Torbjørn Kågen
June 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Preface

This thesis is a diploma study for the Mechanical Engineering master's degree program at the Norwegian University of Science and Technology. The study is written to satisfy the requirements of the subject: TPK4920 Project and quality management, master thesis. The research was conducted in the spring of 2021, from January to June.

We want to thank the companies and informants participating in our case study for sharing your knowledge and experience with us. This study would not have been possible without your contributions.

For invaluable feedback and guidance throughout the study, we would like to thank Assoc. Prof. Bassam Hussein, our supervisor at NTNU. We are grateful for the many hours you have spent helping us maneuver the world of qualitative research.

Lastly, we would like to express our gratitude to Torbjørn Kågen, our external supervisor at Metier OEC AS. Thank you for helping us see how our work can be valuable for practitioners in real-life Norwegian projects.

This study is an original, independent work by the authors Cornelia Reme-Ness and Rebecca Sandtstø, with supervision and guidance from Bassam Hussein and Torbjørn Kågen.

Trondheim, June 10, 2021.

Cornelia Reme-Ness

Rebecca Sandstø

Abstract

Practitioners struggle with a lack of information on how to deal with common challenges of Agile practices within the Norwegian project context.

Our thesis aims to present guidelines for the successful implementation of five selected Agile practices in Norwegian projects. The selected practices are self-organized team, customer involvement, release planning, retrospectives, and stand-up meetings.

Through a case study, we identify common challenges related to the implementation of these selected Agile practices in Norwegian projects and the strategies used to deal with them. Our multiple-case study consists of ten cases investigated through interviews. All cases are Norwegian projects.

In our research, we identify a selection of challenges common across a variety of project types. Furthermore, we describe what strategies project participants use to deal with these challenges and discuss their effectiveness. Conclusively, based on the identified common challenges and their associated strategies, we formulate guidelines for the successful implementation of the selected Agile practices in Norwegian projects.

For practitioners, our study provides guidelines for how to deal with common implementation challenges. Strategies for both the project management and the team are presented. For researchers, our study presents challenges and strategies that need more attention.

Sammendrag

Praktiserende sliter med manglende informasjon om hvordan å håndtere vanlige utfordringer tilknyttet smidige praksiser i en norsk prosjekt-kontekst.

Vår avhandling har som formål å presentere retningslinjer for en vellykket implementering av fem utvalgte smidige praksiser i norske prosjekter. De utvalgte praksisene er selvorganiserende team, kundeinvolvering, leveranseplanlegging, retrospektiver, og stående møter.

Gjennom en case-studie identifiserer vi vanlige utfordringer relatert til implementasjonen av disse utvalgte smidige praksisene i norske prosjekter og strategiene som blir brukt for å håndtere dem. Vår studie består av ti caser, undersøkt gjennom intervjuer. Alle casene er norske prosjekter.

I vår forskning identifiserer vi flere vanlige utfordringer på tvers av ulike prosjekttyper. Videre beskriver vi hvilke strategier prosjektdeltagere bruker for å håndtere disse utfordringene og også deres effektivitet. Avslutningsvis formulerer vi retningslinjer for en vellykket implementering av de utvalgte smidige praksisene i norske prosjekter basert på de identifiserte vanlige utfordringene og deres tilknyttede strategier.

For praktiserende bidrar studien vår med retningslinjer for hvordan å håntere vanlige implementasjonsutfordringer. Strategier for både prosjektledelsen og teamet presenteres. For forskning presenterer studien vår utfordringer og strategier som må vies oppmerksomhet.

Contents

List of Figures

List of Tables

Abbreviations and Terms

Abbreviations

NTNU	Norwegian University of Science and Technology
PM	Project Management

Terms

Agile Manifesto — The Manifesto for Agile Software Development (Beck et al. 2001).

Agile frameworks — A combination of Agile practices to manage a project in line with the Agile Manifesto.

Agile practice — Processes used to guide Agile development in a project.

Backlog — A prioritized list of work that needs to be completed.

Chief Scrum Master — Responsible for setting up the Agile way of working - a role defined in Project 5.

Definition of Done — The acceptance criteria that must be met for a task or feature to be considered completed.

Functional Architect — Responsible for ensuring that the product meets the business' needs.

Iteration planning — A planning meeting held at the beginning of every iteration, also known as sprint planning.

PRINCE2 — A project management methodology and certification.

Scrum — An Agile framework established to handle complex software development.

Scrum Master — A facilitator for Scrum. Not supposed to be a team leader, but rather a Scrum coach for the team.

Sprint — A time unit in the Scrum framework, also known as iteration.

Story Owner — The person responsible for the development of a single user story.

Time-box — Pre-defined time limit - e.g., for sprints or *stand-up meetings*.

User story — A description of a software feature, written from the user's perspective.

1 Introduction

1.1 Background and Motivation

The Agile philosophy emerged as a reaction to the traditional Waterfall model and was first formalized through the formulation of The Manifesto for Agile Software Development by Beck et al. (2001). Its purpose was to establish a framework that could cope with the increasingly rapid changes in technology and business (Dybå & Dingsøyr 2008). Initially, the Agile frameworks were aimed at small, co-located software projects - called *"the Agile sweet spot"* (Kruchten 2004). Presently, they are becoming increasingly popular due to their verified positive impact on project success, also outside the Agile sweet spot (Papatheocharous & Andreou 2014, Serrador & Pinto 2015, Sandstø & Reme-Ness 2021).

The Agile frameworks have also changed the way projects are managed (Cervone 2011). Agile Project Management aims to manage change instead of avoiding it, and this way of managing is therefore useful to projects with a high degree of complexity and uncertainty (Dybå & Dingsøyr 2015, Fernandez & Fernandez 2008). Typical characteristics of Agile Project Management are iterative development cycles, late planning, continuous feedback from customer, collective decision-making, self-management, and a high focus on creativity and learning (Dybå & Dingsøyr 2015, Highsmith & Cockburn 2001, Hoda et al. 2008). Whereas traditional Project Management emphasizes cost, time and scope, the delivery of value to the customer is the highest priority in Agile Project Management (Fernandez & Fernandez 2008). Agile Project Management has also been adopted by widely accepted Project Management methods such as PMBOK (PMI 2017) and PRINCE2 (Metier OEC 2021), through the Agile Practice Guide and PRINCE2 Agile respectively. Furthermore, Agile management roles such as Product Owner and Scrum Master have been introduced (Hoda et al. 2008).

It is common to implement a set of specific Agile practices (also referred to as "practice" from now on) in projects, instead of a complete framework, as addressed by Cao & Ramesh (2008) and Sidky et al. (2007). Nevertheless, the existing body of knowledge lacks knowledge of the implementation of specific Agile practices (Unger-Windeler et al. 2019), the challenges that occur, and available mitigation strategies (Hoda & Murugesan 2016, Nuottila et al. 2016, Masood et al. 2020).

Some research has been conducted on Agile Project Management in Norway - e.g., related to risk management (Siddique & Hussein 2014) and contracting process management (Siddique & Hussein 2017). We identified two studies that targeted the implementation of Agile elements in Norwegian projects: Bernsmed & Jaatun (2019) and Saeeda et al. (2020). Some acknowledged Norwegian researchers have also included Norwegian case studies in their international studies - e.g., Stray et al. (2018).

To the best of our knowledge, no research has been conducted on what the implementation of specific Agile practices entails for Norwegian projects. As the implementation of practices is affected by context and culture (Ayed et al. 2017, Stray et al. 2016), applying conclusions from international research is not forthright. Additionally, we have spoken to practitioners confirming a lack of competence of the implementation of Agile practices in the Norwegian context. Thus, Norwegian practitioners wishing to implement Agile practices in projects have little research to guide them.

We therefore aim to contribute to the existing body of knowledge by establishing guidelines for the successful implementation of common Agile practices in Norwegian projects.

1.2 Research Questions

To establish guidelines for the successful implementation of common Agile practices in Norwegian projects, we will address three research questions through a case study. We will present the common Agile practices we have chosen to research in Subchapter 1.3.

We defined **the first research question** as follows:

> *What are the common challenges of implementing the selected Agile practices in Norwegian projects?*

By investigating the common challenges of selected Agile practices in Norwegian projects, we aim to make practitioners aware of barriers to the implementation of the practices.

We defined **the second research question** as follows:

> *What strategies can be implemented to deal with these common challenges?*

By investigating the strategies for the common challenges addressed by the first research question, we aim to provide practitioners with means to prevent and tackle the challenges that are the most likely to occur.

We defined **the third research question** as follows:

> *How can the selected Agile practices be implemented successfully in Norwegian projects?*

Based on the identified common challenges and their associated strategies (first and second research question), we will establish guidelines for the successful implementation of the selected Agile practices in Norwegian projects.

Through these three research questions, the study will give insight into what is demanded when implementing specific Agile practices and how common challenges can be avoided or mitigated. Hence, we will provide practitioners with the tools to assist a successful implementation of the selected Agile practices in Norwegian projects.

1.3 Scope

Selecting Agile practices to study

We wished to study 3 to 5 common Agile practices, as we considered this to provide an appropriate balance of breadth and debt that is useful to researchers and practitioners. The practices to study were selected in three steps.

First, we combined two sources to identify all relevant practices. VersionOne (2020) reported the most commonly implemented Agile practices globally in 2020. Our last study (Sandstø & Reme-Ness 2021) identified the most commonly reported Agile practices in literature. Combining VersionOne (2020) and Sandstø & Reme-Ness (2021), a list of relevant practices was formed.

Second, we reduced the list based on the availability of a common name. Having shown in our previous study (Sandstø & Reme-Ness 2021) how the same practice often goes under many different names, we considered it important to choose practices with common names. This would increase the probability of identifying all of the available research on the topic.

Third, the availability of literature on the practices was considered important, with ourselves being novices within research.

Supervisors were also consulted as informants of what practices are commonly implemented in Norwegian projects.

The reduction process resulted in the decision to study the following five Agile practices:

- *Self-organized team*
- *Customer involvement*
- *Release planning*
- *Retrospectives*
- *Stand-up meetings*

By aiming our research questions at these Agile practices, we are also able to address the research gaps for *release planning*, *customer involvement* and *stand-up meetings* identified by Heikkilä et al. (2015), Hoda & Murugesan (2016) and Stray et al. (2016) respectively.

Henceforth, we will write the practices under investigation in italic. This is to distinguish between the selected Agile practices and other Agile elements addressed in literature and by informants.

Selecting project types to study

For the identification of related work, reports from any country were considered relevant. Although international research can not be transferred directly to a Norwegian context, the scarcity of literature on the implementation of Agile practices in Norwegian projects made the inclusion of international literature inevitable. Including literature based on project attributes was not deemed relevant nor suitable. We chose to include research done on various project types to ensure that our findings would be applicable to various projects, and to ensure that the identified related work would be based on a sufficient amount of literature.

For the case study, we investigated Norwegian projects. This ensured the inclusion of a Norwegian perspective, thus targeting the identified research gap. As with the literature, we decided to investigate a range of project types. Software and non-software projects, private and public projects, and projects of different sizes were all considered relevant.

1.4 Thesis Outline

Chapter 2, Research Design, describes the methodology used throughout this study. A multiple-case study of ten project cases creates the foundation of this study.

Chapter 3, Case Presentations, introduces the projects of the case study.

Chapter 4 to 8 describe our findings, with one chapter for each selected Agile practice. Each chapter is divided into four subchapters:

1. A short description of the selected Agile practice is provided.
2. The identification of related work summarizes the identified challenges and strategies related to the selected Agile practice.

3. The results from the case study related to the selected Agile practice are presented, project-by-project. Challenges and strategies are identified.
4. A cross-case analysis is performed, and the findings are discussed and connected to the identified related work. The research questions are answered with respect to the selected Agile practice.

Chapter 9, Conclusion, summarizes the main findings regarding the research questions. The answers to the first and second research questions are listed to remind the reader of the findings from Chapter 4 to 8. We also summarize our recommendations for the successful implementation of the selected Agile practices, and organize them based on whom they apply to.

Finally, Chapter 10 presents the limitations of the study, as well as potential areas for further work.

Our study is structured in this somewhat unconventional manner because we consider it to be the most suitable for our study. The structure is preferable for practitioners, as it presents all findings related to a specific practice in one place. For example, if practitioners wish to implement *retrospectives* in their projects, Chapter 7 will provide all information related to *retrospectives*. As the purpose of this study is to contribute to streamlining practitioners' implementation of Agile practices in Norwegian projects, we deemed it valuable using a structure supporting this aim.

2 Research Design

This chapter will describe and justify the chosen research design of the study. In Subchapter 2.1, the overall research design will be presented. The methods to identify the related work will then be presented in Subchapter 2.2, followed by the design of the case study in Subchapter 2.3.

2.1 Overall Research Design

The study aims to address how Norwegian projects can implement the selected Agile practices successfully through the defined research questions in Chapter 1. The study can therefore be classified as explanatory (Marshall & Rossman 2016, p. 78) and is of qualitative nature. Case studies, histories, and experiments are appropriate strategies for explanatory studies (Yin 2003, p. 7). Table 2.1 is based on Yin (2003), page 5, and shows when the various strategies are appropriate.

Strategy	Control of behavioral events	Contemporary events
History	No	No
Experiment	Yes	Yes
Case study	No	Yes

Table 2.1: Research Strategies for Explanatory Studies

As the defined research questions signify the investigation of projects with behaviors outside our control, we considered a case study to be the most suitable strategy for our study. However, it is important to be aware of the fact that it is challenging to conduct good case studies (Yin 2003, p. 11). To ensure we executed a high-quality case study, we established and followed a systematic approach inspired by various sources (Marshall & Rossman 2014, 2016, Yin 2003, Booth et al. 2016, Patton 2002, Brooks et al. 2015).

This study consists of a literature review (to identify related work) and a case study. The overall process can be seen in Figure 2.1. The literature review aimed to uncover the state-of-the-art research on the topic, whereas the case study aimed to investigate the topic further and fill the identified research gaps. Depending on the research design, the literature review can be conducted before, simultaneously with, or after the empirical study. For case studies, the literature review should be performed first (Yin 2003). Hence, after deciding on the overall research design, we commenced the literature review.

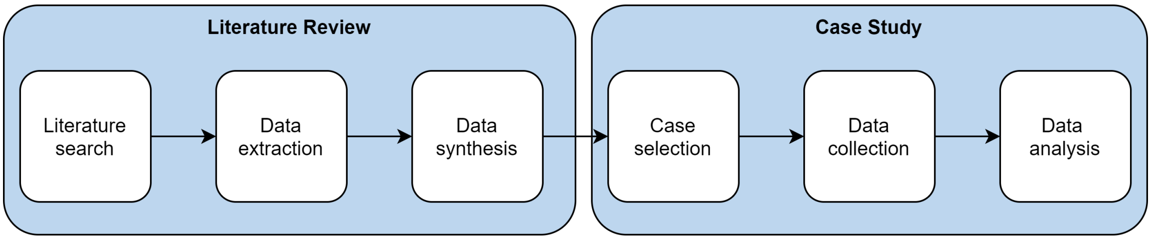


Figure 2.1: Overall Research Design

2.2 Identification of Related Work

To identify related work for our study, we performed a literature review. There are various ways to perform literature reviews. We considered a systematic review to be suitable for this study, as we aim to uncover all evidence on the topic, hence removing researcher bias (Booth et al. 2016, p. 109). However, fully systematic reviews are time-consuming and normally require between 9 and 12 months (Booth et al. 2016, p. 68). Due to the limited time, we could not conduct a fully systematic literature review in addition to the case study. Nonetheless, we have taken inspiration from the systematic approaches presented by Booth et al. (2016) and adopted the elements that we considered beneficial for this study.

2.2.1 Literature Search

The literature search process was conducted in several steps - as illustrated in Figure 2.2. The process was created with adjustments from Booth et al. (2016), pages 110 and 143.

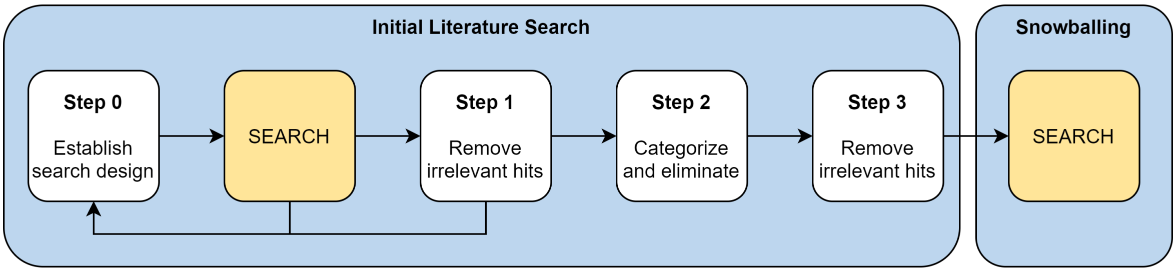
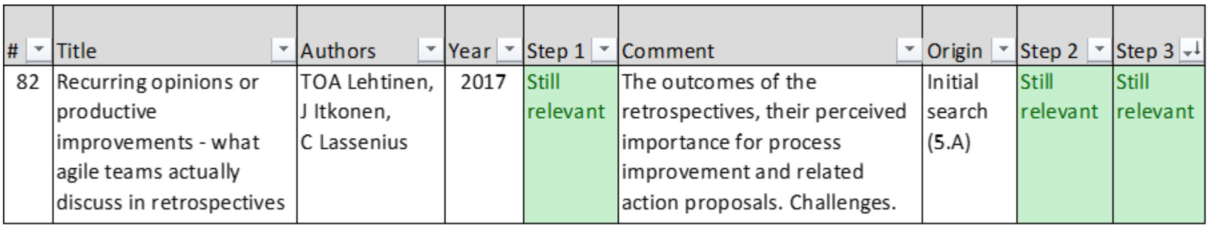


Figure 2.2: Literature Search Process

As shown in Figure 2.2, the process started with the establishment of the search design (Step 0). The search design comprised search criteria, databases to be used, and search phrases. Our Step 0 is what Booth et al. (2016) referred to as "*scoping search*". To establish the search design, we first had to familiarize ourselves with the topic and available literature. It is important to document the literature search process (Booth et al. 2016, p. 110). Thus, we created a log of both the literature search process (in Microsoft Word) and the literature search hits (in Microsoft Excel). In the literature search process log, we documented all iterations of search criteria, databases, and search phrases, along with our own comments and lessons learned about the process and the established search design. In the log of the literature search hits, we documented all possibly relevant hits. We labelled them either as "still relevant" or "discarded" for the various steps of the process, along with the search's origin and our own comments. Figure 2.3 illustrates the log of the literature search hits in Microsoft Excel.



#	Title	Authors	Year	Step 1	Comment	Origin	Step 2	Step 3
82	Recurring opinions or productive improvements - what agile teams actually discuss in retrospectives	TOA Lehtinen, J Itkonen, C Lassenius	2017	Still relevant	The outcomes of the retrospectives, their perceived importance for process improvement and related action proposals. Challenges.	Initial search (5.A)	Still relevant	Still relevant

Figure 2.3: Log of Literature Search Hits

We established the following five search criteria for the literature review:

SC 1	Citations:	5
SC 2	Year:	2001 - 2021
SC 3	Languages:	English
SC 4	Literature type:	Peer-reviewed articles, doctoral dissertations
SC 5	Literature topic:	Related to the defined research questions

SC 1 and SC 4 ensured reliability, whereas SC 2 and SC 5 ensured relevancy. We made exceptions from SC 1 for highly relevant and reliable literature published in 2020 and 2021 to enable the inclusion of important recent findings. Different databases report different numbers of citations. We defined the criterion based on the number of citations reported by the Google Scholar search engine - independent of the used database - to ensure consistency. Literature published prior to 2001 was discarded (SC 2), as most relevant literature on Agile frameworks has been published after the creation of the Agile Manifesto (Dingsøyr et al. 2012). Although We could have included Norwegian literature in the literature review, we limited the search to English publications. We considered the inclusion of literature in Norwegian to be redundant, considering the limited time for this study and the time required to execute the search in two languages.

We used Web of Science and Google Scholar as databases for the literature search. The initial search was performed using Web of Science, as it is an interdisciplinary database with high impact journals (NTNU 2021*b*). The database also enabled us to easily exclude hits not conforming to SC 2, SC 3, and SC 4. The Google Scholar search engine was used for "*snowballing*" - i.e., the process of discovering new literature from the reference list or citations of publications from the initial literature search.

We utilized a variety of search phrases. "Agile" and "project" were included in all search phrases to meet SC 5. For the first research question, we included "challenge" and its synonyms - e.g., "problem" and "issue". For the second research question, we included "strategy" and its synonyms - e.g., "enabler" and "success factor". We also utilized search phrases specifically aimed at the selected Agile practices. An example of a search phrase for the challenges of *stand-up meetings* is:

> agile AND project AND (challenge* OR difficult* OR problem* OR issue*) AND
> (stand-up* OR standup* OR (daily AND meeting*) OR scrum meeting*)

Through Step 1 to 3 in Figure 2.2, we reduced the number of search hits. We assessed validity, reliability, and applicability for all literature (Booth et al. 2016, p. 154).

In Step 1, we removed clearly irrelevant hits. The removal was based on the title and abstract of the search hits within the search criteria, as recommended by Booth et al. (2016).

In Step 2, we categorized the remaining search hits, and then eliminated categories that were outside the scope of this study. Although not described by Booth et al. (2016), we considered this step necessary due to the large amount of remaining hits after Step 1. To identify the categories outside our scope, we used a mind map. Using the mind map, we excluded all search hits that did not assess the implementation of the selected Agile practices directly. The mind map we used can be seen in Appendix B.

In Step 3, we did a quick initial reading of the full text of the remaining hits, followed by the removal of those deemed as irrelevant - also recommended by Booth et al. (2016). The remaining hits after Step 3 were included in the literature review. We used these remaining hits for the snowballing.

During searching and Step 1, we also iterated on the search design, as illustrated by the bottom arrow in Figure 2.2. This is what Booth et al. (2016) referred to as “*revise search strategies*”.

2.2.2 Data Extraction and Synthesis

It is important to establish a synthesis strategy (Booth et al. 2016, p. 224). Figure 2.4 shows the strategy for data extraction and synthesis of this literature review.

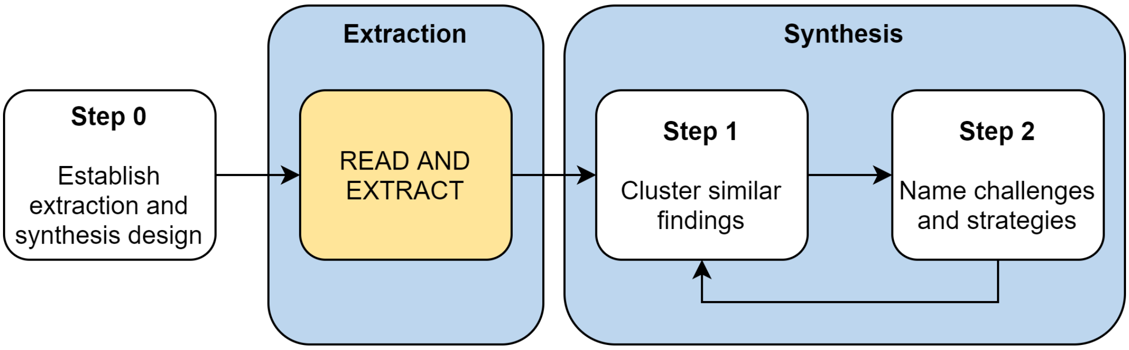
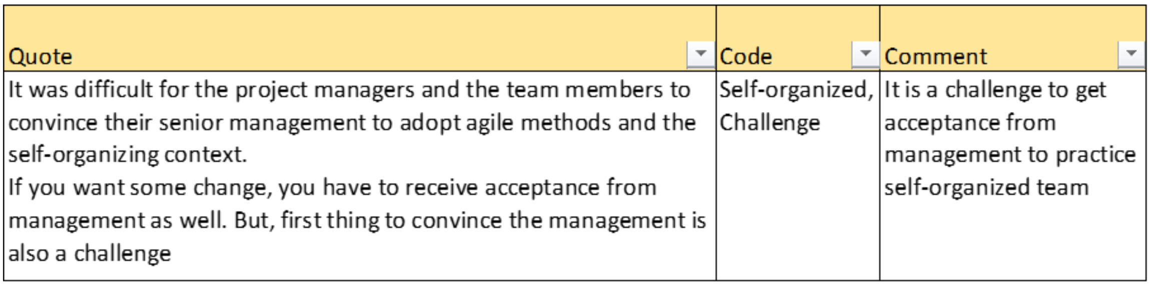


Figure 2.4: Strategy for Literature Extraction and Synthesis

In Step 0 of Figure 2.4, we established the extraction and synthesis design. A consistent data extraction is vital for systematic approaches (Booth et al. 2016, p. 218). Therefore, to ensure consistency in the data extraction of all included literature, we created a codebook in Microsoft Excel. We included all quotes deemed relevant for the research questions in the codebook. We combined each quote with the appropriate pre-established codes, in addition to our own comments - i.e., our short, summarized interpretations of the quotes. We created codes for “challenge”, “strategy”, and each Agile practice. Figure 2.5 shows an example from the codebook, with a quote from Hoda & Murugesan (2016) related to a challenge of *self-organized team*.



Quote	Code	Comment
It was difficult for the project managers and the team members to convince their senior management to adopt agile methods and the self-organizing context. If you want some change, you have to receive acceptance from management as well. But, first thing to convince the management is also a challenge	Self-organized, Challenge	It is a challenge to get acceptance from management to practice self-organized team

Figure 2.5: Example From Codebook

It is important to be able to re-read the original findings when necessary (Booth et al. 2016, p. 219). The combination of quotes and comments in the codebook enabled us to benefit from both our own interpretations and the original findings when synthesizing and analyzing the data. We read each publication in detail before continuing to the next. The reading order was based on considered importance and relevancy.

For the synthesis illustrated in Figure 2.4, we clustered similar findings together (Step 1), and named the various challenges and strategies addressed by literature (Step 2). This was an iterative process, as illustrated by the bottom arrow in Figure 2.4. We considered each interpretation with its corresponding quote as one data point. For each data point, we repeated Step 1 and 2, and the clustering and naming of challenges and strategies were thus iteratively created. Step 2 ensured an explicit connection from the literature review to the research questions defined in Chapter 1 - a vital element of the literature review (Booth et al. 2016, p. 250). The result of the literature review - i.e., the identified related work - is presented for each Agile practice in Chapter 4, 5, 6, 7 and 8.

2.3 Case Study

The case study comprised data collection followed by data analysis. The methods for the collection and analysis of data will be described and justified in this subchapter.

2.3.1 Case Selection

We first had to define the case. The research questions defined in Chapter 1 revolve around Norwegian projects. A single Norwegian project was therefore defined as a single case.

We then had to define the case study design. Case studies can either be single-case or multiple-case. Single-case studies are suitable when testing a well-formulated theory, or when investigating either a unique, typical, revelatory, or longitudinal case (Yin 2003, p. 40-42). However, single-case studies are vulnerable to flaws, as the chosen case might lack the anticipated representativeness. On the other hand, there are multiple-case studies. Although multiple-case studies are more extensive, they are considered more robust (Yin 2003, p. 46). Potentially having several cases reaching the same conclusions under varied circumstances would also expand the external generalizability of our study (Yin 2003, p. 53). To ensure a robust case study, we chose to conduct a multiple-case study comprising multiple Norwegian projects.

We then had to decide on the number of cases to include in the study. With a higher number of cases, the certainty of the results increases (Yin 2003, p. 51). Through the first and second research questions, we aimed at identifying a wide range of challenges and strategies to increase the generalization of the answer to the third research question. Hence, breadth was favoured over depth, making a high number of cases beneficial. Consequently, we chose to include ten Norwegian projects in the study.

2.3.2 Data Collection and Processing

There are six primary data collection methods for case studies (Yin 2003, p. 85): documentation, archival records, interviews, direct observations, participant-observation, and physical artifacts. Each method has its advantages and disadvantages. We considered interviews as the most relevant data collection method for this study, as interviews allow for the targeting of the research questions. Triangulation of the collected data is important to ensure a robust case study (Marshall & Rossman 2016, p. 48), and one way to achieve triangulation is through multiple data collection methods. However, as the implementation challenges and strategies of the Agile practices are subjective to a great extent, we considered interviews to be sufficient as the only data collection method. Therefore, to still ensure triangulation, we utilized several data sources. We aimed for multiple informants from each case and ended up interviewing 22 practitioners in total from the ten Norwegian projects. We also ensured researcher triangulation by both being present at all interviews, with one leading the interviews and the other taking notes.

There are several interview strategies. First, we evaluated focus groups against individual interviews. Focus groups have their advantages but were discarded as they require highly skilled facilitators and are less controllable (Marshall & Rossman 2016, p. 154). Thus, we conducted individual interviews. Second, we evaluated informal interviews, guided interviews, and open-ended interviews. We chose guided interviews, as they allow for the informants to structure their response while still maintaining some structure and consistency (Marshall & Rossman 2016, p. 150).

The interviews were based on an interview guide which we prepared prior to the interviews, and it can be found in Appendix A. The interview guide consisted of four sections: introduction, demographic questions, main interview questions, and ending. In the introduction, we aimed to establish a good relationship with the informant to make the informant feel comfortable sharing his or her views with us (Marshall & Rossman 2016, p. 148). Collecting demographics data in interviews can be time-consuming (Masood et al. 2020) and uninteresting for the informants (Patton 2002, p. 353). Therefore, informants were given a pre-interview questionnaire by e-mail to maximize the time spent on interesting and valuable discussions during the interviews. We created the pre-interview questionnaire according to NTNUs guidelines for the use of questionnaires (NTNU 2021*f*) in UiO Nettskjema (UiO 2021). However, some warm-up questions were still included in the interview in the demographic questions section to make the informants comfortable with the interview format and us. The pre-interview questionnaire also enabled us to target the relevant Agile practices better during the interviews, as the informants' involvement with the selected Agile practices in the projects were known in advance.

The main interview questions were developed and structured based on Patton (2002), pages 348-374, and they provided a direct link to the research questions. The question type was identified for all questions to clarify the anticipated response from the informants (Patton 2002, p. 348). The question type for each question is highlighted in orange in Appendix A. In addition, all questions were open-ended, neutral, singular, and clear, which is a requirement to receive good responses (Patton 2002, p. 353). At the end of the interviews, we ensured that all relevant information had been shared and that the informant was aware of the study's timeline. The duration of each interview ended up being around 1 hour.

It is important to select a sample of informants that benefits the study (Marshall & Rossman 2016, p. 110). The sampling was based on a combination of perceived contribution and convenience, and we favoured breadth over depth. When choosing projects, we targeted Norwegian projects with various case attributes to apprehend the variety of implementation challenges and strategies for the selected Agile practices. When choosing informants, we targeted project resources with various roles - with and without leader responsibilities.

We ensured privacy and safety in the data collection and storage. We filled out the "*Notification form for personal data*" from NSD (2021) and distributed it to all informants prior to the interviews. Through the notification form, we acquired the informants' consent and informed them of the purpose and execution of the study. We also followed the guidelines for data collection by NTNU (2021*a*) and the guidelines for data storage by NTNU (2021*d*) and NTNU (2021*e*). The collected data classified as *"internal"* according to NTNU (2021*c*). Hence, the use of Microsoft Teams for the execution and recording of the interviews (due to COVID-19) and NTNU One Drive for the storage of sensitive information ensured privacy and safety for the informants. All data was deleted when no longer needed.

To ensure a successful data collection, we furthermore prepared ourselves on the cases and execution of interviews prior to the interviews (Yin 2003, p. 57). We utilized publicly available data (e.g., websites) to gain an overview of the projects, in combination with provided data from the informants. We also conducted pilot interviews, as they allowed us to train on the role as interviewers, test the technology, and improve the interview guide (Marshall & Rossman 2016, p. 105). Through transcriptions, interview data was made accessible for preceding processing and analysis (Marshall & Rossman 2016, p. 208). We transcribed using Microsoft Word. Although there exists transcription software (Marshall & Rossman 2016, p. 209), we did not find a transcription software that was able to deal with the various Norwegian dialects.

2.3.3 Data Analysis

We executed preliminary analysis and coding following the steps of template analysis, based on the description by Brooks et al. (2015) and Marshall & Rossman (2014), page 214-233. Template analysis is a method along a continuum between technical standardized methods and immersive, inductive methods (Marshall & Rossman 2014, p. 216). The method is suitable for cross-case analysis. It was chosen due to the use of pre-defined categories, ensuring focus on key areas in our study. Template analysis is also less time-consuming than other methods (Brooks et al. 2015, Marshall & Rossman 2014). This allowed us to analyze a greater quantity of sources than would have been possible with a purely inductive approach.

According to Brooks et al. (2015), the flexibility in the template allows the authors to rush past the descriptive stage. To avoid starting interpretation too early, we performed the initial low-level coding purely descriptive. The initial coding, and later interpretation, was also done by both of us together to avoid a heavy impact of the researchers' bias.

Below is a step-by-step description of the analysis process, based on the steps described by Brooks et al. (2015) and Marshall & Rossman (2014), page 217.

Step 1 - Immersing ourselves in the data

We began the analysis process by getting to know the data and organizing it. By transcribing the interviews ourselves and rereading the transcriptions and interview notes, we immersed ourselves in the data.

Step 2 - Organizing the data

The transcriptions were organized in case hierarchies in Nvivo. The research questions were used as guidelines for the data organization, as suggested by Marshall & Rossman (2014), page 216, and Yin (2003), page 112. Furthermore, we pre-defined categories and subcategories. We defined the categories and subcategories in accordance with our research questions and interview guide. The theory-generated categories were the five Agile practices under investigation:

1. *Self-organized team*
2. *Customer involvement*
3. *Release planning*
4. *Retrospectives*
5. *Stand-up meetings*

The subcategories were the seven aspects of the practices discussed during the interviews:

- Implementation
- Challenges
- Strategies

- Enablers
- Barriers
- Culture
- Contract

Step 3 - Establishing the coding template

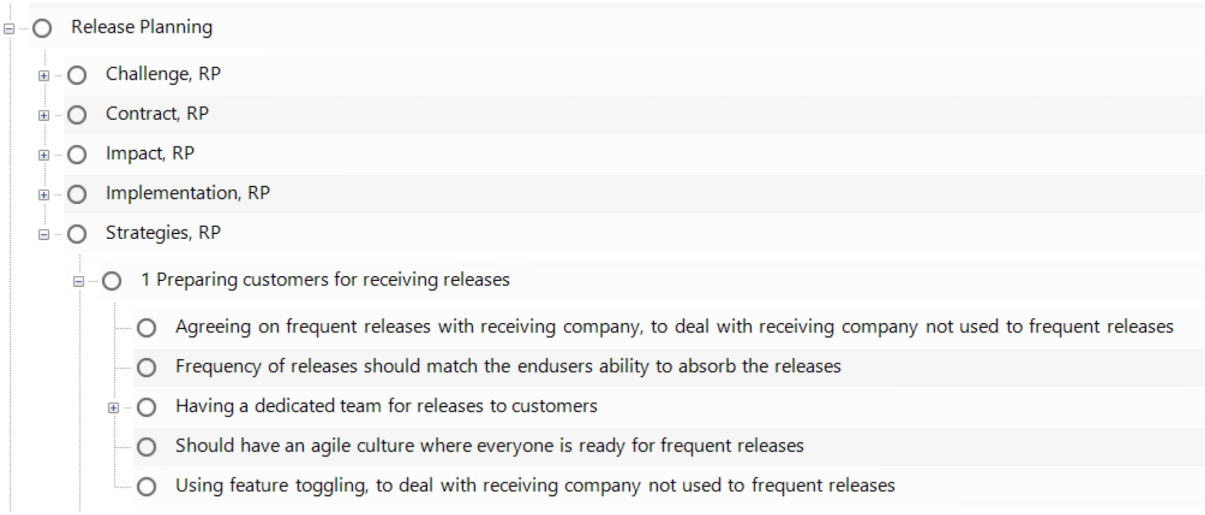
To establish an initial coding template, both of us coded the same transcription. Afterwards, we compared our coding. This led to the creation of the new subcategory “impact”. Even though it was not intended to investigate Agile practices’ impact in our study, all informants touched upon the topic. Additionally, several sub-subcategories for each Agile practice were defined. We deemed it relevant because we asked several specific implementation questions to better understand their challenges and strategies. Therefore, the sub-subcategories would aid in sorting and combining answers to the same questions. To ensure an effective data analysis, detailed bottom-level codes were used so that the transcriptions could be discarded after coding. Namely, codes were detailed enough that there was no need for the transcriptions anymore after coding was completed (NTNU University Library 2021). Strategies for the challenges were coded as children to the challenge they offered a solution to. This also applied to challenges caused by other challenges. This was to ensure traceability so that the connection between different challenges and solutions would be transparent and obvious. Attribute codes - as described by Saldana (2013), page 69-77 - were also defined for the coding template. The attribute codes were defined for two levels; project and individual. Examples of used attribute codes are man-years (project), ownership (project), and role (individual).

Step 4 - Applying the template

We applied the coding template to the rest of the transcriptions. For coding, the transcriptions were switched between authors to code the interviews we had not transcribed. This was to ensure that we both immersed ourselves in all the data. The transcriptions were coded to bottom-level codes created along the way. All bottom-level codes were created as child codes to the top-level pre-defined categories. For example, a bottom-level code could be: “Using feature toggling to deal with receiving company not used to frequent releases”. Throughout the coding, memos were used to note thematic relations between codes and cases. Sub-subcategories were added and altered all the way.

Step 5 - Altering and clustering codes

Upon finishing the initial round of coding, codes were altered once again and clustered together. First, we combined and chose what top-level categories to move forward with. We prioritized the categories based on the criteria from Patton (2002), page 465: *“Evidence showing consistent picture”* and *“the absence of nonassignable cases”*. We disregarded the top-level categories “implementation”, “contract”, and “impact” for further data analysis, as they did not target our defined research questions satisfactory. Second, “barriers” were merged with “challenges”, as “barriers” in this report referred to root causes of challenges and hence were considered “challenges”. Third, “enablers” were renamed “strategies” as it better represented the content coded to this branch. Fourth, codes related to “culture” were combined with and moved to “challenges” and “strategies”, whichever was most suitable. This left us with only “challenges” and “strategies” as relevant top-level categories for each Agile practice. As opposed to the strategies coded under the various challenges, this top-level category comprised strategies not addressing a specific challenge strategies. Lastly, we clustered the detailed codes to create more sub-subcategories. An example of a cluster is shown in Figure 2.6: “1 Preparing customers for receiving releases”. This enabled defining further hierarchical relationships and relations between codes. To relate the results from the case study to the literature review, the clusters similar to the findings from the literature review were identified and renamed where applicable. Both authors performed the altering and clustering together to reduce researchers’ bias.



- ⊟ ○ Release Planning
  - ⊞ ○ Challenge, RP
  - ⊞ ○ Contract, RP
  - ⊞ ○ Impact, RP
  - ⊞ ○ Implementation, RP
  - ⊟ ○ Strategies, RP
    - ⊟ ○ 1 Preparing customers for receiving releases
      - ○ Agreeing on frequent releases with receiving company, to deal with receiving company not used to frequent releases
      - ○ Frequency of releases should match the endusers ability to absorb the releases
      - ⊞ ○ Having a dedicated team for releases to customers
      - ○ Should have an agile culture where everyone is ready for frequent releases
      - ○ Using feature toggling, to deal with receiving company not used to frequent releases

Figure 2.6: Example of Code Structure in Nvivo

Step 6 - Interpreting the data

In multiple-case studies, both single-case analysis and cross-case analysis should be conducted (Yin 2003, p. 50). Figure 2.7 shows the data analysis strategy, with adjustments from Yin (2003), page 50.

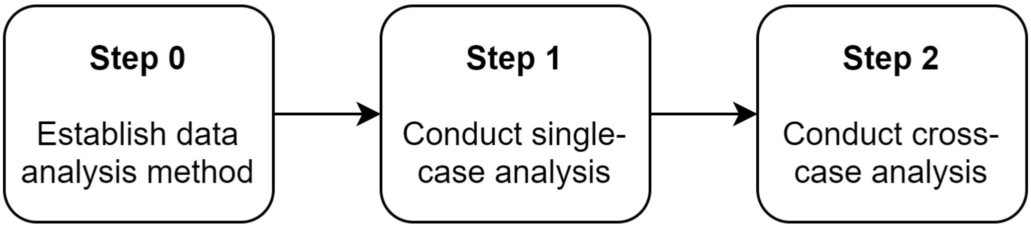

Step 0
Establish data analysis method
Step 1
Conduct single-case analysis
Step 2
Conduct cross-case analysis

Figure 2.7: Strategy for Data Analysis

We established an overall strategy to ensure we treated our evidence fairly (Yin 2003, p. 110-11). Our strategy included the analysis from both a single-case and cross-case perspective, using both quantitative and qualitative approaches, and aided by digital tools. For the single-case analysis, we extracted interview quotes and claims theme-by-theme. We wrote interview notes and thematic memos throughout the process. These were used to add perspectives. To conduct the cross-case analysis, we used a quantitative approach. Quantitative data analysis can be useful when there are numerous cases to be analyzed (Marshall & Rossman 2014, p. 230), (Yin 2003, p. 134). Having data from ten different cases in our study, we deemed a quantitative element useful and reliable. We used built-in features in Nvivo and Microsoft Excel to identify the most common challenges and strategies across the cases under investigation (Marshall & Rossman 2014, p. 226). Identifying claims made in several cases shows there are replications, which again increases the certainty of the results (Yin 2003, p. 51). Credibility in the interpretation was ensured through debriefing with our two supervisors and logging interpretation trails (Marshall & Rossman 2014, p. 230).

2.4 Improvements of Research Design

We suggest the following improvements of the literature review:

- **Include Norwegian publications in search criteria** to target the selected Agile practices in Norwegian projects even better.
- **Utilize more databases in the literature search**, to extend the literature search further, hence enabling even more search hits.
- **Utilize Nvivo for data extraction and synthesis**, as it is a powerful tool that offers even more possibilities than Microsoft Excel.
- **Adjust the search criteria** - e.g., by lowering the citation criterion and allowing the inclusion of non-peer-reviewed articles - to increase the number of references. For some practices, such as *retrospectives*, the availability of relevant articles within our criteria was notably scarce.

We suggest the following improvements of the case study:

- **Triangulate data collection methods**, to increase the robustness of the study. For each project, between three and five informants with different roles was initially desired, as this would ensure the inclusion of various perspectives on the matter. However, due to the availability of informants and the limited time, the informants constituted a limited number of roles for each case.
- **Increase focus on non-occurrences** - both in the interview guide and the data analysis process - to strengthen validity. The interview guide could have been improved by reformulating "What has worked well with the Agile practice?" to obtain fewer replies about impact, and by including an explicit question about successful implementation. We should have spent more time during data analysis searching for negative instances and contrasting findings.
- **Conduct all interviews with a project before moving on to the next project**, to reduce the time spent familiarizing ourselves with each project prior to each interview, and to ensure a better flow in the interviews.
- **Use "member checking"** (Marshall & Rossman 2014) - validate our reasoning with informants - to ensure further credibility of our interpretation.
- **Conduct fewer interviews** to focus more on data analysis. Spending more time addressing topics such as non-occurrences and project attributes could have been beneficial. We conducted 22 interviews, and considering the time we had available, this might have been somewhat excessive.

3 Case Presentations

In this chapter, we will present the case projects to provide the reader with insight into the various aspects of the cases. That way, the reader can get a deeper understanding of the results and discussions of each selected Agile practice (found in Chapter 4, 5, 6, 7 and 8).

Project 1

Project 1 was working on the digitization and modernization of a public register in Norway. The resources worked full-time, and the project consisted of about 80 man-years yearly. The duration of the project was five years, and it finished in December 2020. They worked in sprints - each lasting three weeks - and they had five software development teams with a team size of 8-12 people.

We interviewed three informants from the project; a Development Manager, a Developer, and a Product Owner / Functional Architect.

Project 2

Project 2 was also working on the digitization and modernization of a public register in Norway, focusing on processing and storing input data. The resources worked full-time, and the project consisted of about 85 man-years yearly. The planned duration was seven years, with the finish date in 2023. The development was structured in sprints of three weeks. They had four software development teams - each consisting of 7-9 team members.

We interviewed two informants from the project; a Development Coordinator, and a Team Manager.

Project 3

Project 3 was one of the projects that stood out. They conducted a preliminary study on how to best digitize and modernize yet another public register in Norway. No software development was performed, and neither the team manager nor the team members had any previous experience with Agile frameworks or Agile practices. Less than five man-years were involved yearly. The project resources worked part-time, and they constituted one team. The duration was three years, with a planned ending in 2021, and the sprints lasted 2 weeks.

We interviewed two informants from the project; a Project Manager / Scrum Master, and a Team Member.

Project 4

Project 4 was working on digitizing and modernizing a part of the same public register in Norway as Project 2, though focusing on making the data available for the user. The customer representatives were working part-time, whereas the remaining resources worked full-time. They had 2-week sprints and about ten man-years yearly. There was one team of 12-13 people.

We interviewed one informant from the project; a Team Manager / Deployment Coordinator.

Project 5

Project 5 was owned by a private oil and gas company in Norway. The project aimed to digitize a work process. The project consisted of seven developer teams, with 2-6 people in each team.

In some teams, all project resources worked full-time, whereas other teams had a mix of part-time and full-time resources. There were 25 man-years involved yearly. Each sprint lasted two weeks, and the total project duration was two years. The project was going to end in 2022.

We interviewed one informant from the project; a Chief Scrum Master.

Project 6

Project 6 was also privately owned - but by a Norwegian technology company. This project also stood out, as their delivery was a product consisting of both software and hardware solutions. The planned finish date was in 2021, with a total duration of two years, and they used 2-week sprints. They had about 25 man-years yearly, divided in two Agile teams and 3-4 other teams, all consisting of 2-7 people. The teams were fixed company teams assigned to several projects, thus answering to several Project Managers. They were not specifically created for the project, nor would they break up after the ending of the project. The teams therefore worked part-time in the case project.

We interviewed two informants from the project; a Project Manager, and a Tester.

Project 7

Project 7 was also working on the digitization and modernization of a public register in Norway - just like Project 1, 2 and 4. The duration was seven years, and the finish date was in 2021. Each sprint lasted three weeks. The project was fairly large, with about 150 man-years yearly. There were six Scrum teams with maximum nine people, and the resources worked full-time in the project. Resources from several companies were involved.

We interviewed two informants from the project; a Contract Responsible / Organization Developer, and an Assistant Project Manager.

Project 8

Project 8 was owned by the same private oil and gas company in Norway as Project 5. The delivery of the project was web applications for the digitization of work processes. The project lasted four years and ended in 2020. There were about 80 man-years involved yearly, distributed in six teams. The work was divided in sprints of two weeks, and the resources worked full-time.

We interviewed one informant from the project; a Project Manager.

Project 9

Project 9 worked on the digitization and modernization of a public health sector service. It consisted of less than five man-years yearly, and they constituted one team. As with Project 6, the team was a fixed company team assigned to several projects, and the resources were therefore part-time involved in the case project. Each sprint lasted three weeks.

We interviewed four informants from the project; a Project Manager, a Functional Architect, a Team Tester / Test Manager, and a Team Manager / Developer.

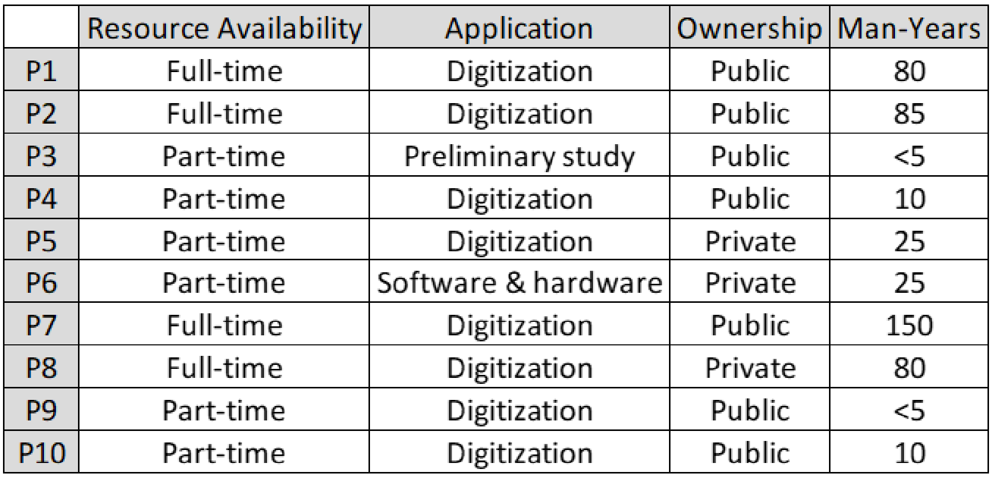
Project 10

Project 10 worked on implementing new functionality and changes to a digital public register in Norway with interfaces to other registers. The project had a planned finish date in 2022, with a total duration of three years. About 10 man-years were involved yearly. All resources were organized in one team. This team varied for each sprint, as many resources alternated between working in the project and in operation. Many resources therefore worked part-time in the project. The work was divided in sprints of three weeks. The project was also especially limited on time and cost.

We interviewed four informants from the project; a Project Manager, a Tech Lead / Lead Developer, a Test Manager, and a Product Owner / Scrum Master.

Overview of Main Case Attributes

The main attributes of the various case projects are summarized in Figure 3.1. We would like to highlight that projects with a substantial number of part-time resources were classified as "part-time", even though some full-time resources might have been involved.



	Resource Availability	Application	Ownership	Man-Years
P1	Full-time	Digitization	Public	80
P2	Full-time	Digitization	Public	85
P3	Part-time	Preliminary study	Public	<5
P4	Part-time	Digitization	Public	10
P5	Part-time	Digitization	Private	25
P6	Part-time	Software & hardware	Private	25
P7	Full-time	Digitization	Public	150
P8	Full-time	Digitization	Private	80
P9	Part-time	Digitization	Public	<5
P10	Part-time	Digitization	Public	10

Figure 3.1: Main Attributes of Case Projects

4 Agile Practice: Self-Organized Team

This chapter will target the three research questions defined in Chapter 1 for the selected Agile practice *self-organized team*, following the research design described in Chapter 2. As we have chosen a nontraditional report structure that we consider to be beneficial for this study, we will present all related work, results, and discussion of *self-organized team* in this chapter. To make sure the reader understands the Agile practice under investigation in this chapter, we will commence with a short description of *self-organized team*.

4.1 Description of Practice

Self-organized team is a complex practice consisting of various elements. Among these elements are self-assignment (Masood et al. 2020, Hoda & Murugesan 2016), cross-functionality (Hoda & Murugesan 2016), autonomy (Werder & Maedche 2018, Hoda & Murugesan 2016), and collective estimations (Hoda & Murugesan 2016). Some of the responsibilities of the *self-organized team* in projects are requirements management, estimation, change request management, and clarifications (Hoda & Murugesan 2016).

In our study, we treat self-organization as a continuum. In one end, we find fully *self-organized teams*. These are teams without a traditional leader, and that define their own work and timelines. Typically, these can make decisions with economic consequences and are multidisciplinary enough to solve all tasks within the team. Achieving a completely *self-organized team* is challenging and requires the adaption of organizational structure and culture (Kakar 2017). On the other end of the continuum, you have teams that approximate traditional teams. These are teams with little authority of how to distribute and solve their tasks. We have included responses from teams all along the continuum of *self-organized team*. Only the project that did not implement *self-organized team* at all (Project 3) was not taken into account.

In our interviews, we treated "autonomous team" as a synonym of *self-organized team*. Some interviewees were more familiar with "autonomous" than with "self-organized", and vice versa. We did not deem it necessary for our study to separate these two terms.

4.2 Related Work

In this subchapter, literature related to *self-organized team* and the research questions will be presented. We have identified the related work by following the methods described in Subchapter 2.2.

4.2.1 Challenges

Challenges of *self-organized team* have been addressed by Masood et al. (2020), Hoda & Murugesan (2016), Hoda (2011), Werder & Maedche (2018), Moe et al. (2012) and Drury et al. (2012). These are listed in Table 4.1. The most frequently cited challenges are considered more applicable and robust and will therefore be discussed. The remaining challenges are considered self-explanatory and will not be discussed in further detail.

C1: Dealing with external dependencies

External dependencies can limit the *self-organized team* (Masood et al. 2020, Hoda & Murugesan 2016). One such external dependency is the dependency to managers. Team managers can

ID	Challenge	References
C1	Dealing with external dependencies	Masood et al. (2020) Hoda & Murugesan (2016)
C2	Lacking understanding of practice	Hoda (2011) Werder & Maedche (2018) Masood et al. (2020) Hoda & Murugesan (2016)
C3	Lacking focus on task dependencies	Hoda & Murugesan (2016) Masood et al. (2020)
C4	Lacking resources and competence	Hoda & Murugesan (2016) Moe et al. (2012) Masood et al. (2020)
C5	Working in distributed teams	Masood et al. (2020)
C6	Lacking team orientation	Moe et al. (2012) Drury et al. (2012)
C7	Lacking clear direction	Moe et al. (2012)
C8	Having passive team members	Moe et al. (2012)
C9	Making inappropriate decisions	Moe et al. (2012) Drury et al. (2012)

Table 4.1: Challenges of *Self-Organized Team* From Literature

hinder the *self-organized team* in different ways - consciously or unconsciously (Masood et al. 2020, Hoda & Murugesan 2016). By giving suggestions, they advocate their own way of doing things, making the task distribution less self-assigned (Masood et al. 2020). Masood et al. (2020) argued that some team managers found it difficult to adjust to not assigning tasks, as they were so used to doing it. Project Managers can be sceptic to the practice (Masood et al. 2020), possibly because they perceive it to be risky (Hoda & Murugesan 2016). Consequently, they can decide to reject the implementation of *self-organized team* completely, or they can officially accept it but still act in a way that does not align or enable the practice. This causes challenges to emerge (Hoda & Murugesan 2016). Project Managers that still expect heavy documentation and the same, traditional kind of reporting hinder the freedom that is necessary for the team to become a *self-organized team.*

Another such external dependency is the dependency to the customer. Direct and frequent interaction with the customer combined with customer's changing requirements cause challenges for the *self-organized team* in projects (Hoda & Murugesan 2016). The customer can wish to alter the requirements at any time during the project, and it can be difficult for the team to reject these change requests (Hoda & Murugesan 2016). Accepting change requests results in a constantly changing backlog, which is challenging for the *self-organized team* (Masood et al. 2020), as it means redoing estimations or even canceling sprints in the more extreme cases (Hoda & Murugesan 2016). The changing requirements from the customer can be a consequence of a lack of common product vision (Hoda & Murugesan 2016).

Lastly, a person outside the team can give urgent work to the team - e.g., the Project Manager or the customer. Masood et al. (2020) described urgent work as *"one of the most influential factors that constrain practicing self-assignment"*. Thus, urgent work is a great challenge for the *self-organized team.* The urgent work restricts the freedom of the *self-organized team* and the available time and resources in the sprint (Masood et al. 2020).

C2: Lacking understanding of practice

Team members can have trouble understanding the role and responsibilities associated with the *self-organized team* (Hoda 2011, Werder & Maedche 2018, Masood et al. 2020, Hoda & Murugesan 2016). Expectations of the *self-organized team* can be unclear (Werder & Maedche 2018), new team members are unfamiliar with how *self-organized team* is practiced in that particular team (Masood et al. 2020), and some team members are unfamiliar with the practice in general (Hoda & Murugesan 2016). Team members who are completely new to the practice can continue working in traditional ways, or they can struggle to embrace the new freedom (Hoda & Murugesan 2016, Masood et al. 2020).

C3: Lacking focus on task dependencies

Lack of focus on dependencies between tasks poses challenges for the *self-organized team* (Hoda & Murugesan 2016, Masood et al. 2020). When team members self-assign dependent tasks oblivious of the dependencies, the progress is affected (Masood et al. 2020). According to Hoda & Murugesan (2016), this lack of focus derives from the lack of assigned responsibility for identifying, monitoring, and resolving dependencies.

C4: Lacking resources and competence

Lack of resources and competence hinders self-assignment, thus hindering the implementation of *self-organized team* (Masood et al. 2020). This happens because the lack of resources and competence limits the freedom to choose tasks, resulting in an unequal distribution of tasks. Hoda & Murugesan (2016) argued that it was challenging to achieve the necessary cross-functionality. Although both self-assignment and cross-functionality are considered vital elements of the *self-organized team*, self-assignment can threaten the cross-functionality, as team members have the freedom to only or mostly select comfortable tasks that require no learning and growth (Hoda & Murugesan 2016). Achieving cross-functionality implies team members executing tasks outside their area of expertise, which naturally is more time-consuming and can cause delays (Masood et al. 2020). On the other hand, lack of cross-functionality reduces the team members' ability to participate in the decisions and helping each other (Moe et al. 2012).

C6: Lacking team orientation

Lack of team orientation poses challenges for the *self-organized team* (Moe et al. 2012, Drury et al. 2012). This can result in some team members making decisions without involving the others (C9), as addressed by Moe et al. (2012). Another consequence can be that competence is not passed on from senior to junior resources (Drury et al. 2012), hence limiting the competence building (C4).

C9: Making inappropriate decisions

The *self-organized team* can reach inappropriate decisions (Moe et al. 2012, Drury et al. 2012). Drury et al. (2012) reasoned that the input from the experienced resources can play a smaller role as the entire team is supposed to make decisions collectively, whereas Moe et al. (2012) claimed that the challenge could be caused by lack of team orientation (C6).

4.2.2 Strategies

Strategies for *self-organized team* have been addressed by Stray et al. (2016), Masood et al. (2020), Hoda & Murugesan (2016), Werder & Maedche (2018), Crowston et al. (2007) and Babb et al. (2014). These are listed in Table 4.2. The IDs in parentheses show what challenges each strategy deals with, and those without parenthesis are enablers of the practice in general and do not target any specific challenges. The most frequently cited strategies are considered more applicable and robust and will therefore be discussed. The remaining strategies are considered self-explanatory and will not be discussed in further detail.

Strategy	References
Having a skilled facilitator (C1, C2, C4, C8)	Stray et al. (2016) Masood et al. (2020) Hoda & Murugesan (2016) Werder & Maedche (2018) Crowston et al. (2007)
Shielding the team (C1)	Hoda & Murugesan (2016)
Keeping backlog consistent and well-groomed	Masood et al. (2020)
Implementing sprint rule	Masood et al. (2020)
Utilizing tools (C5)	Masood et al. (2020) Hoda & Murugesan (2016)
Assigning tasks without team manager (C1)	Masood et al. (2020)
Acquiring management support (C1)	Werder & Maedche (2018) Masood et al. (2020) Hoda & Murugesan (2016)
Assigning team members to several projects (C4)	Werder & Maedche (2018)
Hiring external consultants (C4)	Werder & Maedche (2018)
Defining roles and responsibilities clearly (C1, C2)	Hoda & Murugesan (2016) Werder & Maedche (2018)
Highlighting task dependencies (C3)	Masood et al. (2020) Hoda & Murugesan (2016)
Creating independent tasks (C3)	Masood et al. (2020) Hoda & Murugesan (2016)
Isolating dependent tasks across sprints (C3)	Masood et al. (2020)
Having flexible estimates (C4)	Masood et al. (2020)
Training (C2, C4)	Masood et al. (2020) Hoda & Murugesan (2016)
Controlling the quality of executed tasks (C4)	Crowston et al. (2007)
Providing enough task information (C2)	Masood et al. (2020) Crowston et al. (2007)
Building appropriate culture (C8)	Masood et al. (2020) Hoda & Murugesan (2016) Stray et al. (2016)
Ensuring understanding of product and stakeholders (C1)	Masood et al. (2020) Werder & Maedche (2018) Hoda & Murugesan (2016)
Establishing common team goals	Werder & Maedche (2018)
Ensuring effective communication and feedback	Hoda & Murugesan (2016) Werder & Maedche (2018)
Implementing sprints	Werder & Maedche (2018)
Utilizing reflective practices	Babb et al. (2014)

Table 4.2: Strategies for *Self-Organized Team* From Literature

Having a skilled facilitator (C1, C2, C4, C8)

The implementation of *self-organized team* depends on the team manager and how he or she leads the team (Stray et al. 2016, Masood et al. 2020). The team manager should encourage, guide, ensure knowledge-sharing, value teams and empower autonomy, according to Masood et al. (2020). Hoda & Murugesan (2016) argued that the team manager also should be able to constantly adapt to the context by balancing the necessary intervention in every situation. Furthermore, it is vital to establish trust between the team manager and the team members, as team members must be given the freedom to make mistakes and learn from them (Hoda & Murugesan 2016, Werder & Maedche 2018). Masood et al. (2020) highlighted this as a suitable strategy for the challenges related to external dependencies (C1), lack of understanding of the practice (C2), lack of resources and competence (C4), and passive team members (C8). As a way to deal with urgent work, the team manager can provide necessary information regarding task necessities and dependencies, and thereafter encourage the *self-organized team* to volunteer and self-assign the tasks, (Crowston et al. 2007, Masood et al. 2020, Werder & Maedche 2018). To ensure cross-functionality, the team manager can oversee that the self-assignment decisions enable learning and growth for all team members (Masood et al. 2020). When dealing with shy or less-confident team members, the team manager can do one-on-one mentoring (Masood et al. 2020). It is also important that the team manager is able to balance cross-functionality and learning (Masood et al. 2020, Hoda & Murugesan 2016); The productivity of the team is considerably affected with an excessive focus on learning and growth, whereas limiting the learning and growth also limits the cross-functionality (Masood et al. 2020).

Acquiring management support (C1)

Support from project management or organizational management is essential for the successful implementation of *self-organized team* (Werder & Maedche 2018). This implies giving their verbal commitment and financial support, aligning their actions with their verbal commitment and the practice, and clearly stating their expectations towards the team. To deal with managers that are sceptic or have rejected the implementation of *self-organized team* (C1), the team can provide evidence of a successful implementation of *self-organized team* (Masood et al. 2020, Hoda & Murugesan 2016). Masood et al. (2020) suggested implementing the practice on a part of the project initially, thus convincing management through showing.

Building appropriate culture (C8)

A certain culture must be built to facilitate the implementation of *self-organized team* (Masood et al. 2020). There must be a culture for learning and improvement (Stray et al. 2016, Hoda & Murugesan 2016). For the shy or less-confident team members (C8), it is important that the culture advocates open discussions (Masood et al. 2020). The team members must also have a positive attitude towards the practice (Masood et al. 2020).

Ensuring understanding of product and stakeholders (C1)

Understanding the product and stakeholders is a strategy that enables *self-organized team* (Masood et al. 2020, Werder & Maedche 2018). The team should have easy access to users and good knowledge about them, according to (Werder & Maedche 2018), indicating that a successful implementation of *customer involvement* facilitates *self-organized team.* If the team lacks the required competence and experience to execute the necessary user research (C4), external consultants can be hired to ensure sufficient understanding of the product (Werder & Maedche 2018). As lack of common product vision can result in frequently changing requirements (C1), establishing a common product vision between the *self-organized team* and the customer (Hoda & Murugesan 2016) can be beneficial. This vision can be established through, e.g., *release planning*, and it should span several iterations.

4.3 Results

In this subchapter, the results from the case study related to *self-organized team* will be presented and discussed. We will identify challenges and associated strategies, and the IDs in parentheses behind the strategies show what challenges they deal with. The results from each project will be addressed separately.

Project 1

In Project 1, there was a lack of focus on task dependencies, as explained by I9:

> Because each team was allowed to kind of do things their own way. And if we were to clarify everything in advance, it would have taken longer. So that was probably something one chooses to do. [...] And then you have to accept that some solutions start to deviate from each other, and that can make it more difficult to upgrade it [the solution] afterwards. (I9, P1)

Hence, not addressing the task dependencies can lead to deviating solutions that are difficult to upgrade later. On the other hand, addressing the task dependencies can be time-consuming.

Conclusively, the following challenges were addressed in interviews with Project 1:

- C3: Lacking focus on task dependencies

No strategies were suggested in interviews with Project 1.

Project 2

Two informants shared how external dependencies can limit the *self-organized team.* I12 found it challenging to solve tasks with dependencies outside the team:

> As always, it's about having access to things outside the team. That's what's difficult. Things that we can solve within the team mostly go well. [...] If we need clarifications on some business-specific topic, then there are dependencies outside the team. And that is kind of what is nearly always the most challenging to solve. Because then you are dependent on the priorities of that gang. (I12, P2)

This resulted in reduced flexibility for the *self-organized team.*

We also found that some were struggling to understand what was expected of them in a *self-organized team.*

Conclusively, the following challenges were addressed in interviews with Project 2:

- C1: Dealing with external dependencies
- C2: Lacking understanding of practice

No strategies were suggested in interviews with Project 2.

Project 3

No challenges or strategies were identified in Project 3.

Project 4

Project management can inflict time-consuming tasks on the *self-organized team.* In Project 4, the project management required more statistics on the time spent on the various tasks, even though the informant believed the *self-organized team* functioned well without it. Such demands limit the freedom of the *self-organized team.*

I5 explained that some personalities could dominate the team, making the team less self-organized:

> But there are especially some that demand a lot of space, that I think might have directed too much. So that has been unfortunate. And it might have subdued others a little bit. (I5, P4)

Thus, the result can be a team organized and directed by one or more strong personalities instead of by the entire team.

The project also struggled to agree on the product within the team. The solution was to divide the team into two sub-groups with different focus areas. This worked well.

Responsibilities could become pulverized, as the team was expected to deal with HR cases within the team without anyone being responsible for aiding them on how to do it. The strategy for the challenge was to give feedback to management.

From Project 4, we found that it was important that team members knew their roles well. Many of the team members had not received sufficient training in their role in the *self-organized team.*

As a reply to how the *self-organized team* could have worked better in the project, I5 insinuated a lack of resources and shared the importance of having a full-time team:

> But then, maybe the most important thing is that they [team members] are 100% dedicated to the team and do not have tasks on the side. (I5, P4)

Conclusively, the following challenges were addressed in interviews with Project 4:

- C1: Dealing with external dependencies
- C10: Dominating personalities
- C11: Reaching agreements within the team
- C12: Pulverizing responsibilities
- C2: Lacking understanding of practice
- C4: Lacking resources and competence

The following strategies were suggested in interviews with Project 4:

- Dividing into smaller teams (C11)
- Giving feedback to managers (C12)
- Training (C2)
- Having full-time resources (C4)

Project 5

External dependencies in Project 5 hindered the implementation of *self-organized team.* The informant suggested that the organization's tradition for strict control of risk and cost limited self-organization.

I7 explained that the responsibilities were pulverized in the *self-organized team*:

> You could say that the responsibilities get sort of pulverized. Everyone in the team has some responsibilities for the ceremonies, and everyone in the team has some responsibilities for the backlogs. But then you could say that there are no one who really owns it. (I7, P5)

Pulverized responsibilities thus cause lack of ownership. The project was going to implement several strategies. They wanted to combine several small teams into one bigger team. They also wanted to properly define Scrum Masters and Product Owners and use external consultants as Scrum Masters. By clearly defining the roles and responsibilities, they would also address the lack of understanding of the responsibilities associated with the roles, as people were struggling to really own and undertake their roles.

Conclusively, the following challenges were addressed in interviews with Project 5:

- C1: Dealing with external dependencies
- C12: Pulverizing responsibilities
- C2: Lacking understanding of practice

The following strategies were suggested in interviews with Project 5:

- Combining into bigger teams (C12)
- Defining roles and responsibilities clearly (C12, C2)
- Hiring external consultants (C12)

Project 6

I13 explained that the project struggled to align the project management with the *self-organized team*:

> When you're a Project Manager, you're often historically in control. And to then let go a little bit, maybe without having the framework to say that "This is where we're going, but you can prioritize". So I would say that's a challenge, the transition to becoming self-organized. Because it requires a lot from the rest of the organization, that we let go and give them [the team] the control. (I13, P6)

Another informant also shared that some team members were not available to be full members of the *self-organized team*; They had been asked from a higher level to focus on certain delayed tasks and not to participate in all team meetings, such as iteration planning and *retrospectives*, hence limiting the self-organization. Furthermore, the rigid organizational culture made the team members passive, and it was considered challenging to be a *self-organized team* with passive team members.

There was also a lack of understanding of expectations of the team members in a *self-organized team*, as described by I8:

> I believe the biggest hinder is that people simply are not aware of what is expected of them. (I8, P6)

The team was told to become a *self-organized team* without any guidance on how to do it in practice. Many had no experience working in a *self-organized team*, and they struggled to get themselves involved. Clearly expressing role expectations to the team members was suggested as a strategy.

Lastly, the project lacked resources and competence. Specific knowledge vital to several teams was only possessed by one team member in one of the teams. Having some team members only as part-time members of the *self-organized team* also created a frustrating situation.

Conclusively, the following challenges were addressed in interviews with Project 6:

- C1: Dealing with external dependencies
- C2: Lacking understanding of practice
- C4: Lacking resources and competence

The following strategies were suggested in interviews with Project 6:

- Defining roles and responsibilities clearly (C2)

Project 7

From Project 7, we found that it can be challenging to acquire the right competence in the *self-organized team*. When asked about the right composition of competence, I14 highlighted a mix of developers and field representatives from the customer, and also a mix of seniority:

> But you should probably have a mix of juniors and seniors. Because juniors might bring a fresher culture than certain seniors. So it's important to have a competence mix, because juniors can also challenge more established seniors on several things. And it's important that seniors train juniors as the next generation. So I believe it's a mix of things. (I14, P7)

Conclusively, the following challenges were addressed in interviews with Project 7:

- C4: Lacking resources and competence

The following strategies were suggested in interviews with Project 7:

- Acquiring a mix of competence (C4)

Project 8

I11 explained that they struggled with a lack of understanding of *self-organized team*:

> I think I underestimated how much communication is required for people to understand their roles. So even though we were good at writing down and have written roles and responsibilities, it wasn't enough. People were constantly asking and were uncertain of who should do what. (I11, P8)

A suggested strategy was to spend more time having plenary discussions of roles and responsibilities.

Conclusively, the following challenges were addressed in interviews with Project 8:

- C2: Lacking understanding of practice

The following strategies were suggested in interviews with Project 8:

- Defining roles and responsibilities clearly (C2)

Project 9

Project 9 struggled with external dependencies limiting the *self-organized team.* The project was governed by a project model that inflicted restrictions on the team. It was also challenging that the rest of the organization did not align well with the *self-organized team*, as shared by I17:

> And we sometimes see that it's a challenge to have a Project Manager as an intermediary. Because that means it's automatically a form of waterfall outside the team. So it's completely fine that the team is autonomous, but the biggest challenge is that everyone on the outside are not autonomous. That they always come in and decide. That makes the team feel that they're not autonomous. They don't have the influence that you actually want. (I17, P9)

Thus, the organization's structure with the Project Manager as an intermediary did not align well with the *self-organized team.* It was therefore suggested to remove the Project Manager. We believe it is important to highlight that the *self-organized team* was a fixed team within the organization (as explained in Chapter 3) involved in various projects with various Project Managers acting as intermediaries between the *self-organized team* and the customer. Therefore, we would argue that the removal of the intermediate Project Managers can work well as a strategy in similar organizations, but we would not recommend it in general.

Project 9 struggled with a lack of competence, as the *self-organized team* was fairly small. They were expected to execute certain tasks without having been given resources with the appropriate competence required to execute these tasks.

Conclusively, the following challenges were addressed in interviews with Project 9:

- C1: Dealing with external dependencies
- C4: Lacking resources and competence

The following strategies were suggested in interviews with Project 9:

- Removing intermediaries (C1)

Project 10

Project 10 sometimes struggled to reach agreements within the team. This was a challenge related to decisions made early on in the project.

I22 explained that the non-Agile organization surrounding the project limited the *self-organized team*:

> I would say that the organization I'm in now is not very agile on any level. It's a very, very traditional and static organization. So it's like you almost have to look for doors that are slightly open and elbow your way through, within the room for maneuver that we have. (I22, P10)

New project resources were not properly followed up, resulting in lack of understanding of how to work in a *self-organized team.* This applied especially to the hired consultants. The project solved this by assigning mentors to all new project resources.

As many as three informants from Project 10 described the lack of resources and competence. The challenge occurred due to limited money in the project. The lack of resources and competence resulted in chaos for the developers. In Project 10, many resources alternated between working in the project and in operation (as explained in Chapter 3) - i.e., the available project resources varied. It was challenging when resources with critical roles were absent, as shared by I21:

> So there are some resources that are more burdened than others. And when they have a critical role, and we are faced with a situation or a meeting or a discussion where that role has to state something, it has been challenging. Then, we often have to say that "Today, you take that hat instead", and then we move on. (I21, P10)

Hence, they used rotating roles as a strategy for the challenge - where the role is referred to as a *"hat"*. Another informant said the strategy was not ideal, but it had worked better than expected. The project also hired extra external resources as a way to fill the lack of resources and competence. However, as the project was public, the flexibility was limited by various frame agreements. It was therefore considered more challenging to hire external consultants in this public project than in private projects.

Conclusively, the following challenges were addressed in interviews with Project 10:

- C11: Reaching agreements within the team
- C1: Dealing with external dependencies
- C2: Lacking understanding of practice
- C4: Lacking resources and competence

The following strategies were suggested in interviews with Project 10:

- Assigning mentors (C2)
- Rotating roles (C4)
- Hiring external consultants (C4)

4.4 Discussion

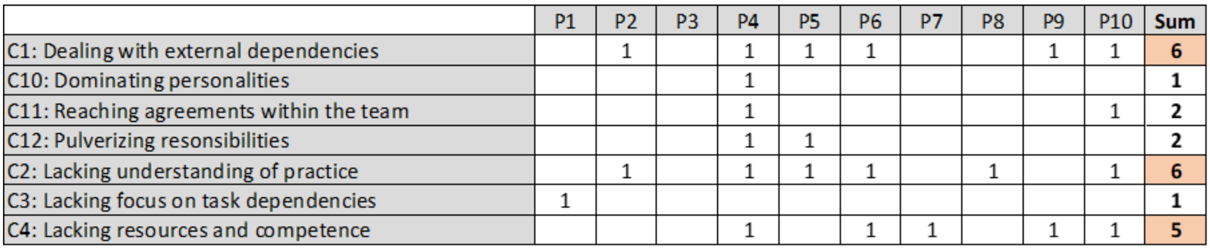
In this subchapter, we will conduct a cross-case analysis where we discuss our findings and connect them to the identified related work. We will also answer our three research questions with respect to *self-organized team.*

Common challenges of *self-organized team* in Norwegian projects

We identified various challenges in interviews with the ten case projects. Figure 4.1 shows the number of projects where the challenges have been identified. Challenges found in three or more projects are considered common. The sum is yellow for challenges identified in 3-4 projects (none for *self-organized team*), and red for those identified in five or more. The remaining challenges - with white sums - are considered uncommon. We will only discuss the common challenges.

The two most common challenges - reported by six projects - dealt with external dependencies (C1) and the lack of understanding of *self-organized team* (C2).

The challenge of dealing with external dependencies (C1) was addressed in interviews with nine informants. The challenge has also been identified in literature (Masood et al. 2020, Hoda &



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Sum
C1: Dealing with external dependencies		1		1	1	1			1	1	**6**
C10: Dominating personalities				1							**1**
C11: Reaching agreements within the team				1						1	**2**
C12: Pulverizing resonsibilities				1	1						**2**
C2: Lacking understanding of practice		1		1	1	1		1		1	**6**
C3: Lacking focus on task dependencies	1										**1**
C4: Lacking resources and competence				1		1	1		1	1	**5**

Figure 4.1: Challenges of *Self-Organized Team* From the Case Study

Murugesan 2016). The external dependencies can be related to tasks with dependencies outside the team (Project 2), project management (Project 4 and 6), or the interface between the project and the organization (Project 5, 6, 9 and 10). It is more challenging to solve tasks with dependencies outside the team (Project 2). The project management can inflict tasks on the team that limits its self-organization (Project 4 and 6), or it can be challenging for the project management to give control to the *self-organized team* (Project 6). The organization can have established ways of doing things defined by culture (Project 5, 6 and 10) or be structured in a way that does not align well with the *self-organized team* in the project (Project 9).

The lack of understanding of *self-organized team* (C2) was addressed in interviews with six informants. Not everyone understands what it means to be self-organized (Project 2 and 10), and people are unaware or uncertain of the responsibilities associated with the various roles in the *self-organized team* (Project 4, 5, 6, and 8). There seems to be a lack of training (Project 4), follow-up (Project 10) and clear expectations (Project 5, 6, and 8). Solely distributing written expectations regarding roles and responsibilities is not sufficient (Project 8). The lack of understanding results in people struggling to undertake their roles (Project 5 and 8) and getting themselves involved in the *self-organized team* (Project 6). The challenge has also been addressed by Hoda (2011), Werder & Maedche (2018), Masood et al. (2020) and Hoda & Murugesan (2016).

The last of the common challenges was the lack of resources and competence (C4) - identified in five projects. It is considered difficult to acquire the right composition of competence for the *self-organized team* (Project 7). Lack of competence and resources make it difficult to fill the roles of the *self-organized team* and achieve the necessary cross-functionality and autonomy (Project 6 and 9). Therefore, we are not surprised that it can be challenging to have part-time resources in a *self-organized team* (Project 4, 6, and 10), considering they are not as available to fill the various roles and participate in the collective decision-making as the full-time resources. The lack of resources and competence result from cost limitations in the projects (Project 10), as both resources and competence cost. Hoda & Murugesan (2016), Moe et al. (2012), Masood et al. (2020) and Drury et al. (2012) all found this challenge in their studies.

Consequently, the common challenges of *self-organized team* in Norwegian projects are the following:

- C1: Dealing with external dependencies
- C2: Lacking understanding of practice
- C4: Lacking resources and competence

Strategies for the common challenges

As external dependencies limit the self-organization of the team (C1), removing unnecessary intermediaries can be a good strategy. In certain organizational structures surrounding the projects, the Project Manager can be perceived as an intermediary (Project 9). However, Project Managers are not unnecessary intermediaries for most projects, and careful assessment should back up the decision to remove an intermediary. We argue that the most important thing is to ensure a seamless interface between the *self-organized team* and its surroundings, and removing unnecessary intermediaries can be a way of doing that.

Clearly defining roles and responsibilities can increase the understanding of how to work as a *self-organized team* (C2). The strategy was suggested in interviews with Project 5, 6, and 8, thus highlighting its importance. Solely defining the roles and responsibilities written is not sufficient, and communicating and discussing the roles and responsibilities verbally several times with all project resources should be prioritized (Project 8). Hoda & Murugesan (2016) and Werder & Maedche (2018) confirm the strategy, and they also recommend using it to deal with external dependencies (C1).

Training is a strategy to increase the understanding of the roles in the *self-organized team* and the associated responsibilities (C2) - as suggested in Project 4. Masood et al. (2020) and Hoda & Murugesan (2016) also suggested to train the team members as a strategy for the lack of understanding of the practice, but also as a strategy for the lack of resources and competence (C4).

Assigning mentors to new project resources can increase their understanding of how to work in a *self-organized team* (C2). The strategy was implemented in Project 10, and we believe it is a good way of utilizing the more experienced resources to teach and train the new ones.

Acquiring full-time resources for the *self-organized team* can be used as a strategy for the lack of resources and competence (C4) - quite contradictory to the strategy of assigning resources to several projects (Werder & Maedche 2018). The acquisition of full-time resources was suggested by an informant from Project 4, which was a project with a mix of full-time and part-time resources. Although not mentioned as a strategy in Project 6, part-time resources created a frustrating situation, indicating that having full-time resources could have been a suitable strategy. However, the challenge related to part-time resources only emerged in practice in Project 4, 6 and 10, with the common denominator being resources available at different times or to various extents. We therefore reason that having resources in the *self-organized team* that are available at the same time and to the same extent is a valuable strategy.

Acquiring resources with a mix of competence can be a good strategy to ensure a beneficial composition of competence (C4). This builds upon the cross-functionality of the *self-organized team* (Hoda & Murugesan 2016). The resources should preferably be a mix of seniority, and a mix of developers and field representatives from customer (Project 7).

Rotating the roles of the *self-organized team* can be a strategy for the lack of resources and competence (C4). Although it can work satisfactorily in practice (Project 10), we believe the rotation of roles is a way of accepting and dealing with the challenge after it has occurred, more than solving the actual problem. We therefore recommend rotating roles only if there are no available alternatives to increase the resources or competence.

The resources and competence in the *self-organized team* can also be increased (C4) by hiring external consultants. This strategy is also confirmed by Werder & Maedche (2018). Both Project 5 and 10 explicitly used the strategy. However, it can be challenging for public projects to hire external consultants, as they can have less flexibility to involve additional resources when needed than private projects (Project 10).

We thus recommend the strategies presented in Table 4.3 for the common challenges of *self-organized team* in Norwegian projects.

Challenge	Strategy
C1: Dealing with external dependencies	Removing intermediaries
C2: Lacking understanding of practice	Defining roles and responsibilities clearly Training Assigning mentors
C4: Lacking resources and competence	Having full-time resources Having equally available resources Acquiring a mix of competence Rotating roles Hiring external consultants

Table 4.3: Strategies for *Self-Organized Team* From the Case Study

Successful implementation of *self-organized team* in Norwegian projects

Based on the interviews and our own reflections, we recommend the following implementation of *self-organized team* in Norwegian projects:

- Ensure a seamless interface between the project and its surroundings - e.g., by removing unnecessary intermediaries.
- Define roles and responsibilities of the *self-organized team* clearly and communicate them both written and verbally to all project resources many times. **Avoid** solely defining roles and responsibilities written, or assuming people will easily undertake their roles without sufficient guidance and communication.
- Train project resources in how to work in a *self-organized team.*
- Assign mentors to the new project resources to ensure exchange of experience of the roles and responsibilities in the *self-organized team.*
- Consider acquiring full-time resources, as there are challenges related to having part-time resources. If using part-time resources; Ensure that they are available at the same time and to the same extent. **Avoid** involving resources with differing availability in the project, as this can create frustration and difficulties in filling the roles of the *self-organized team.*
- Acquire resources with a mix of competence - e.g., a mix of seniority and from various stakeholders.
- Rotate the roles if there are no alternatives to increase the resources and competence. **Avoid** rotating the roles unless strictly necessary.
- Hire external consultants to increase the number of resources and the competence in the *self-organized team.*

5 Agile Practice: Customer Involvement

This chapter will target the three research questions defined in Chapter 1 for the selected Agile practice *customer involvement*, following the research design described in Chapter 2. As we have chosen a nontraditional report structure that we consider to be beneficial for this study, we will present all related work, results, and discussion of *customer involvement* in this chapter. To make sure the reader understands the Agile practice under investigation in this chapter, we will commence with a short description of *customer involvement*.

5.1 Description of Practice

Customer involvement is a crucial practice for most Agile teams (Hoda et al. 2011). Customers in Agile projects are often expected to write user stories (Mohammadi et al. 2008) - a description of a software feature, written from the user's perspective. Furthermore, the customer is often expected to participate at demos - a meeting held at the end of each iteration to demonstrate the current project progress to the stakeholders.

Sometimes, the customer also takes on the role of a Product Owner (Vanhala & Kasurinen 2019). A "Product Owner" is a role used in many Agile frameworks. The Product Owner is responsible for the backlog and the prioritization and maximization of stakeholders' value. Another role discussed in this chapter is "customer proxy" - a person chosen to represent the customer when "real" customer representatives are not available. In our study, we define the "customer representative" as a person from the customer's organization that represents the customer's interest. The availability of the customer representative can vary from only being available by e-mail to being engaged in the project full-time. Lastly, we use "customer resources" about people from the customer who interact in some way - actively or passively - with the project team. Customer proxies, customer representatives and Product Owners are all considered customer resources.

Throughout this study, we have chosen to consider the user as a type of customer. We deemed it suitable for two reasons: First, it made it easier for the informants to answer our questions. Second, we believe it makes it easier for the reader to follow our reasoning - also for case studies with complex customer-supplier structures.

5.2 Related Work

In this subchapter, we will present literature related to *customer involvement* and the research questions. We have identified the related work by following the methods described in Subchapter 2.2.

5.2.1 Challenges

Challenges of *customer involvement* have been addressed by Hoda et al. (2011), Hoda & Murugesan (2016), Tanner & Mackinnon (2015), Pikkarainen et al. (2008) and Drury et al. (2012). These are listed in Table 5.1. The most frequently cited challenges are considered more applicable and robust and will therefore be discussed. The remaining challenges are considered self-explanatory and will not be discussed in further detail.

ID	Challenge	References
C1	Dealing with external dependencies	Hoda et al. (2011)
C2	Lacking understanding of practice	Hoda & Murugesan (2016) Hoda et al. (2011)
C5	Working in distributed teams	Hoda et al. (2011)
C13	Lacking clear requirements and needs	Hoda & Murugesan (2016) Pikkarainen et al. (2008) Tanner & Mackinnon (2015) Drury et al. (2012)
C14	Lacking sufficient customer involvement	Hoda et al. (2011)

Table 5.1: Challenges of *Customer Involvement* From Literature

C2: Lacking understanding of practice

The customer can lack the understanding of how to practice *customer involvement* in Agile projects (Hoda & Murugesan 2016, Hoda et al. 2011). Customers can impose their traditional ways of working on the Agile team and thus not act as Agile customers (Hoda et al. 2011). This is especially the case for large customers.

C13: Lacking clear requirements and needs

Both developers and customers experience lack of comprehension of the requirements (Tanner & Mackinnon 2015); Developers struggle to understand the customer's requirements (Pikkarainen et al. 2008), and customers fail to define clear and high-quality requirements with associated acceptance criteria (Hoda & Murugesan 2016, Tanner & Mackinnon 2015).

The Agile frameworks welcome changing requirements, but also assume they are introduced during iteration planning or iteration review (Tanner & Mackinnon 2015). Customer changing requirements more frequently than once every iteration is a challenge for Agile development (Hoda & Murugesan 2016, Tanner & Mackinnon 2015), possibly due to customers failing to understand and define iteration goals (Tanner & Mackinnon 2015).

When multiple customers are involved, they can also struggle to agree on common requirements and needs, as they all have different priorities and wishes (Drury et al. 2012). This makes it challenging for the developers to know what to prioritize.

5.2.2 Strategies

Strategies for *customer involvement* have been addressed by Hoda et al. (2011), Cao et al. (2009), Layman et al. (2006), Tanner & Mackinnon (2015) and Hoda & Murugesan (2016). These are listed in Table 5.2. The IDs in parentheses show what challenges each strategy deals with, and those without parenthesis are enablers of the practice in general and do not target any specific challenges. The most frequently cited strategies are considered more applicable and robust and will therefore be discussed. The remaining strategies are considered self-explanatory and will not be discussed in further detail.

Using customer proxy (C13, C14)

Using a customer proxy or a team of customer proxies to play the role of the customer can be useful (Cao et al. 2009, Hoda et al. 2011, Layman et al. 2006). Having the technical manager or the Project Manager act as the customer could have a positive effect (Cao et al. 2009, Hoda

Strategy	References
Training (C2)	Hoda et al. (2011)
Using customer proxy (C13, C14)	Cao et al. (2009) Hoda et al. (2011) Layman et al. (2006)
Collaborating on backlog grooming (C13)	Tanner & Mackinnon (2015)
Having frequent in-depth discussions (C13)	Tanner & Mackinnon (2015)
Shielding the team (C13)	Hoda & Murugesan (2016)
Adapting contract (C1)	Hoda et al. (2011)
Prioritizing based on the readiness of user story (C13)	Hoda et al. (2011)
Clarifying level of customer participation up-front (C14)	Hoda et al. (2011)
Using story owners (C2)	Hoda et al. (2011)
Using demos (C2, C5)	Hoda et al. (2011)

Table 5.2: Strategies for *Customer Involvement* From Literature

et al. 2011). Layman et al. (2006) further specified that the person to play the customer role must be chosen before the project start, have the knowledge to make decisions regarding scope and functionality, be available for the team, and be invested in the project.

5.3 Results

In this subchapter, the results from the case study related to *customer involvement* will be presented and discussed. We will identify challenges and associated strategies, and the IDs in parentheses behind the strategies show what challenges they deal with. The results from each project will be addressed separately.

Project 1

Project 1 was ordered and executed by the same organization. They struggled to identify the customer. To acquire the customer aspect, they introduced executive officers from the organization's operation and Product Owners. Whereas the executive officers were passively available as customer representatives, the Product Owners were actively involved in the project and organized in a customer proxy team.

There was a lack of sufficient involvement from the customer. This problem was often not discovered until after the project. Having more customer representatives involved full-time in the project was suggested as a solution. Having customer representatives available by phone or in workshops was - according to I9 - not optimal:

> So there have been many hired consultants in this project. I would like it if there were more actual executive officers from customer active in the project. Not that you only talk to them on the phone or talk to them in a workshop every third week. Maybe things would have been different. But yes ... We managed well enough. (I9, P1)

Conclusively, the following challenges were addressed in interviews with Project 1:

- C15: Struggling to identify customer
- C14: Lacking sufficient customer involvement

The following strategies were suggested in interviews with Project 1:

- Using customer proxy (C15)
- Having full-time customer resources (C14)

Project 2

Project 2 was ordered and managed by the same organization. To define customer needs and product requirements clearly, the project used a customer proxy team. The team consisted of employees from the organization's operation. However, the project struggled with the limited capacity of the customer proxy team. Having less capacity in the customer proxy team than in the developer team led to delays in clarifications. Developers would frequently ask for more resources on the customer proxy team to attempt to improve the situation, but their wishes were not fulfilled.

The culture difference between the customer and developer teams can pose challenges. I12 described how this could make communication difficult:

> That's always what's the challenge, the communication. Isn't it? [...] Yes, we speak a bit techy and a bit of our own language, and they speak in their language, and then we don't understand each other completely. (I12 P2)

Thus, abbreviations and subject terminology complicated the communication between developers and customer. The establishment of a terminology guide was mentioned as an implemented strategy. Nevertheless, the most widely used strategy by project participants was simply consulting colleagues.

Conclusively, the following challenges were addressed in interviews with Project 2:

- C13: Lacking clear requirements and needs
- C14: Lacking sufficient customer involvement
- C16: Mixing work cultures

The following strategies were suggested in interviews with Project 2:

- Using customer proxy (C13)
- Increasing the number of customer resources (C14)
- Establishing terminology guide (C16)
- Consulting colleagues (C16)

Project 3

Project 3 was ordered and executed by the same organization. Nevertheless, the department receiving the project's product was completely separate from the project team. Therefore, they functioned as an external customer. No specific customer resources had been appointed to the project as customer representatives, and as a result, all employees of the receiving department were considered customers.

Both informants expressed their frustration with the customers' lack of interest in participating and giving feedback. Their demos functioned only as progress presentations, almost completely without feedback from the customers present. Some of the problem might be explained by the technical complexity of the presentations, preventing the customer representatives from participating due to a lack of understanding of the subject. Lack of dedicated customer representatives lead to confusion of what customers to reach out to, which also contributed to the problem. I3 shared the challenge related to lack of feedback and customer participation:

> What we want in those meetings [demos], is getting feedback from our customer's employees. They have a lot of knowledge of what is demanded out in the field. But what one experiences is that we deliver a lot of information and ask if everyone has anything to say. And then nothing comes, not a single peep. That's very hard. I feel like that's the challenge, to get others involved". (I3, P3)

The organizational culture hindered *customer involvement* in the project. The customers were not used to working in an Agile way and seemed to lack understanding of the importance of their feedback. Responding time was long and the customers normally worked on long-term projects with deliveries that were thoroughly worked through before each release. As a result, the releases were too abstract for the customers, making it challenging for them to give feedback.

Conclusively, the following challenges were addressed in interviews with Project 3:

- C14: Lacking sufficient customer involvement
- C2: Lacking understanding of practice
- C17: Struggling with Agile releases

No strategies were suggested in interviews with Project 3.

Project 4

Project 4 was ordered and managed by the same organization, and they had customer representatives from operation involved in the project as part-time project resources. The quality of the requirements from the customer was a challenge in Project 4. This was connected to customer resources struggling to find time to participate in meetings. I5 linked this challenge to the fact that these customer representatives were part-time involved in the project. When we asked if I5 suggested to have full-time customer resources as a solution to this challenge, I5 confirmed:

> Yes, absolutely. And also with regard to what they deliver, like the quality of requirements and such. It would probably be very different if they had enough time to work on it. (I5, P4)

Conclusively, the following challenges were addressed in interviews with Project 4:

- C13: Lacking clear requirements and needs
- C14: Lacking sufficient customer involvement

The following strategies were suggested in interviews with Project 4:

- Having full-time customer resources (C13, C14)

Project 5

Project 5 was ordered and executed by the same organization, and they had all internal employees from the receiving department functioning as customers. However, they were not involved as project resources and were therefore considered as external customers.

We found that the project lacked sufficient *customer involvement*. The customers were busy, thus limiting the available time to give feedback to the project. This challenge was also connected to the customers' lack of understanding of their responsibilities in the project, as explained by I7:

> I think that the challenges might also be for the customers to understand what responsibilities they have. If you know what I mean? That it's one thing to attend demos, but we want them to have an opinion and to also take ownership of the implementation and extract value and such. (I7, P5)

To solve the lack of sufficient *customer involvement*, they introduced a focus on only initiating a project if a specific customer representative would say, "Yes, I want this, and I am going to follow it up". Furthermore, they trained the managers of the receiving department in the value of early feedback, so that they could also influence their employees positively and increase their understanding of their responsibilities as customers, hence increasing their involvement. The project also had performance reviews where they evaluated all added value. In these reviews, the customer had to answer for the added value, and not the development team. This underlined the customer's responsibility to implement releases and take them into use.

Another challenge of *customer involvement* was that the customers were used to seldom and large releases of finished products. The customers were not used to frequently giving feedback on incomplete products - which is required in Agile projects. They did not want to take incomplete products into use, as they did not see its value. The unwillingness to take frequent releases into use was especially great in high-risk projects. To meet this challenge, they always evaluated the risk before implementing releases. Where the risk was considered to be high, the incomplete products were only used in operation as a supplement.

The project also struggled with customers not knowing what they wanted. The problem was considered common by the informant.

Lastly, there was a lack of trust between customer and supplier. Traditionally, the customers were sceptical of the suppliers and afraid they would charge them more money than necessary. They worked on this inherent scepticism in the project, but it still emerged as an issue from time to time.

Conclusively, the following challenges were addressed in interviews with Project 5:

- C14: Lacking sufficient customer involvement
- C2: Lacking understanding of practice
- C17: Struggling with Agile releases
- C13: Lacking clear requirements and needs
- C18: Lacking trust between stakeholders

The following strategies were suggested in interviews with Project 5:

- Appointing specific individuals as customer resources (C14)
- Training (C2)
- Arranging performance reviews (C2)
- Evaluating risk (C17)

Project 6

No challenges or strategies were identified in Project 6.

Project 7

Project 7 was a project ordered and managed by the same organization. They struggled to identify the customer, as they had a large selection of end-user groups. An informant described the identification of customer as their greatest challenge of *customer involvement*. They used a customer proxy team to identify the customer. The customer proxy team consisted of executive officers representing the end-user groups, and they were a part of the project. Ideally, the actual end-users should sit alongside the developers, as shared by I10:

> But the closer to the end-user, the better. Like, it's better for the one developing solutions, I would say. In a fully Agile world, the developers sit together with the people who are gonna use the product, right. And they decide together and have fun. But it's not always like that. (I10, P7)

However, using a customer proxy team appeared to be sufficiently effective in their project. In addition, meetings were held to identify suitable people to answer the different categories of questions. Where no one could be identified, they established task forces to find a solution.

The customers lacked a sufficient understanding of the practice. An informant described the customers' inexperience in how to work in a project as the greatest challenge. The customer resources needed training in both project methodology and specification of requirements. Therefore, they used a designated project introduction course as a strategy. Further, team managers and process managers were trusted to train the customer resources further. It was found that "specialist training" was best done by the project resources authorizing those roles. A substantial part of the training was also done by trying and failing. Despite the efforts invested in training, I14 described that the project management concluded that the training was an area of improvement:

> We have now reached the point where we're writing an experience report on the totality. And regarding the training of field experts from customer, we see that we simply could have been better at this. (I14, P10)

This might indicate that training project participants in Agile roles is more complicated than first assumed.

Conclusively, the following challenges were addressed in interviews with Project 7:

- C15: Struggling to identify customer
- C2: Lacking understanding of practice

The following strategies were suggested in interviews with Project 7:

- Using customer proxy (C15)
- Establishing task forces (C15)
- Training (C2)

Project 8

Project 8 was ordered and executed by the same organization. They struggled to identify the customer and to achieve sufficient involvement, as explained by I11:

> That's also a thing we have used a bit of time on and still don't think we are very good at, like who the customer is, having the customer's perspective. [...] And I

> wished we had more end-user involvement. That may be something you can never get enough of, I think. (I11, P8)

The lack of clear customer and insufficient involvement resulted in little focus on customer value. The culture in Project 8 did not help either, as their work culture was simply more centered around getting the job done than on what would create value for customer. To establish a clear customer, they assigned resources from operation in the organization as Product Owners. These Product Owners were specialists highly skilled in their respective fields, and they were a part of the project. This gave the developers a direct connection to the customer and someone who knew the end user. The strategy was considered partly successful, but did not fully solve their problem. Although not implemented due to lack of competence of *customer involvement* and Agile frameworks in general, more user testing - also early in the project - was suggested as a strategy for the insufficient *customer involvement*. Thus, the lack of understanding of the practice hindered the implementation of user tests as a strategy.

Balancing the developers' contact with the customer and shielding them from managerial tasks was a challenge. An informant shared that while the developers should thoroughly understand the project domain, their time should not be wasted in time-consuming meetings with the customer - e.g., conducting root cause analysis or value stream analysis. They established no overall project strategy to cope with this challenge; Every Product Owner decided themselves how to balance these aspects.

Filling the Product Owner role was described as challenging, as the role should have the mandate to prioritize product functions, while at the same time collecting opinions from various stakeholders. Thus, the Product Owner needs to be comfortable having the last say, even when their decision leaves some stakeholders dissatisfied. The informant shared that some Product Owners chose to withdraw and take decisions independently without involving other stakeholders, resulting in sub-optimal decisions. This finding indicates that personal skills should be taken into account when choosing a Product Owner.

Conclusively, the following challenges were addressed in interviews with Project 8:

- C15: Struggling to identify customer
- C14: Lacking sufficient customer involvement
- C2: Lacking understanding of practice
- C19: Balancing developers' customer contact
- C20: Filling Product Owner role

The following strategies were suggested in interviews with Project 8:

- Appointing specific individuals as customer resources (C15)
- Conducting more user tests (C14)
- Appointing adequate Product Owner (C20)

Project 9

Project 9 consisted of a developer team that delivered to a customer. The developer team and the customer belonged to separate organizations, and no one from the customer organization was involved as project resources. Instead, one of the team members from the executing organization acted as Product Owner.

It was challenging to establish trust between the developer team and the customer. The customer would, for example, complain over high time estimates on tasks. The lack of trust was not perceived as a severe challenge, and only one out of four informants mentioned it as a challenge.

The absence of customer resources in the team was a challenge. I16 - the Product Owner - preferred having resources from the customer in the project team, especially since they were dealing with a complex solution:

> But it's a pretty complex solution. Both technical, as I've understood it, even though I don't fully understand that part. But also the functional needs. So I think that it would have been beneficial sometimes to ... shall I dare to say ... that they [the customers] sat in the team while we developed tasks for them. Because that is a practice that I'm very used to from [previous employee], that they own the needs themselves, so they are a representative. Sitting in the team and can make decisions continuously. (I16, P9)

However, the remaining three informants regarded the *customer involvement* as sufficient. I18 even said that they "definitely had enough customer involvement" - in clear contrast to the statement of I16 above. Their success occurred despite not having any customer resources in the project team. Part of the success was explained by the project team and customer knowing each other well. A weekly project meeting with customer was also considered an important contributor. The team members' active participation in discussions with the customer impacted positively. Lastly, having a common digital platform (Slack) enabled quick clarifications of doubts developers had regarding product requirements.

Conclusively, the following challenges were addressed in interviews with Project 9:

- C18: Lacking trust between stakeholders
- C14: Lacking sufficient customer involvement

The following strategies were suggested in interviews with Project 9:

- Having customer resources as part of the project team (C14)
- Establishing close communication between customer and team members (C14)

Project 10

Project 10 had many stakeholders; The project was ordered by one public agency, paid for by a second agency, and fulfilled the requirements of a third. Consequently, the project struggled to identify their customer. To define a customer, they chose to involve representatives from important user groups in the project.

Finding the right decision-maker among stakeholders was also challenging, making the customer's needs and requirements unclear. Getting an opinion on a function was easy enough, but finding who could make the decision to implement it was more difficult. An informant indicated that they possibly should have given more authority to the project team to cope with the unclear requirements and needs.

Working distributed impacted the *customer involvement* negatively. The threshold for communication between the customer representatives and the developer team increased. This led to more formal meetings and the loss of frequent, informal clarifications. An informant concluded that working distributed demanded more from the project culture, indicating that distributed teams need more team building than co-located ones.

Three of the informants claimed that customers never were involved enough. Increased customer involvement was considered necessary to improve and optimize the project. A suggested solution was to include more hours for the customer representatives in the time budget. However, this is not a simple matter, and someone must be willing to take that extra cost. I22 proposed the strategy of having a systematic approach to *customer involvement* with an establishment of specific customer resources before starting the project:

> First, you need to know who the users are and what user groups you have for what you are making or developing. And then, you must find "do we have any actual users or a user group who represent those roles that you have defined". And if these are someone who one can use throughout the development. So everything, from the start really, with specification of requirements and design as well. Someone you can ask questions or who can clarify. And then, user testing and participating during implementation, when it's set. (I22, P10)

To compensate for the customer representatives' lack of availability, bypassing them and speaking directly to the end-users was also suggested. By streamlining the communication through skipping an intermediary, the project saves a little bit of time. An informant underlined that they implemented this strategy because the project team knew the end-users from before. Moreover, we discovered that appointing an adequate customer representative is important - especially if customer resources are scarce. A customer representative with long experience in the customer organization was considered appropriate, as the project had experienced that inexperienced customer representatives could lead to suboptimal products. Additionally, the representative should have experience from most of the departments that receive the project's product. That way, more of the user groups would be represented in the project - with fewer resources.

Conclusively, the following challenges were addressed in interviews with Project 10:

- C15: Struggling to identify customer
- C13: Lacking clear requirements and needs
- C5: Working in distributed teams
- C14: Lacking sufficient customer involvement

The following strategies were suggested in interviews with Project 10:

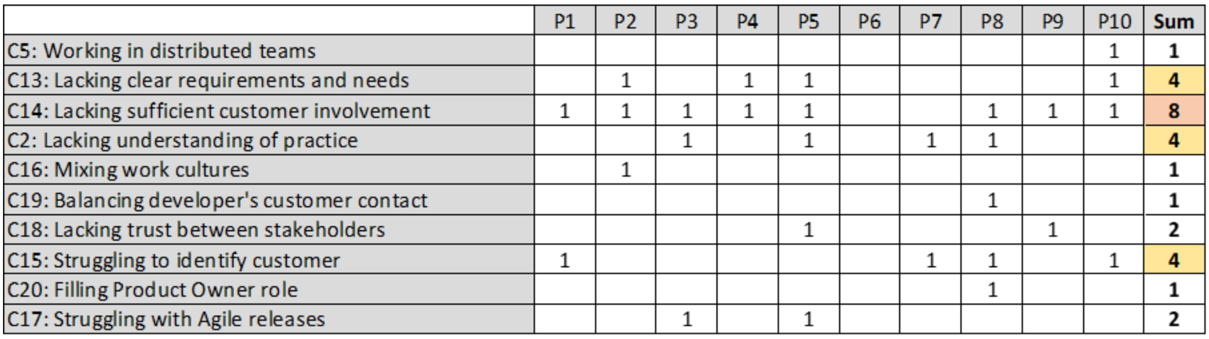
- Increasing the authority of project teams (C13)
- Increasing the number of customer resources (C14)
- Appointing specific individuals as customer resources (C14, C15)
- Removing intermediaries (C14)

5.4 Discussion

In this subchapter, we will conduct a cross-case analysis where we discuss our findings and connect them to the identified related work. We will also answer our three research questions with respect to *customer involvement*.

Common challenges of *customer involvement* in Norwegian projects

We identified various challenges in interviews with the ten case projects. Figure 5.1 shows the number of projects where the challenges have been identified. Challenges found in three or more projects are considered common. The sum is yellow for challenges identified in 3-4 projects and red for those identified in five or more. The remaining challenges - with white sums - are considered uncommon. We will only discuss the common challenges.



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Sum
C5: Working in distributed teams										1	1
C13: Lacking clear requirements and needs		1		1	1					1	4
C14: Lacking sufficient customer involvement	1	1	1	1	1			1	1	1	8
C2: Lacking understanding of practice			1		1		1	1			4
C16: Mixing work cultures		1									1
C19: Balancing developer's customer contact								1			1
C18: Lacking trust between stakeholders					1				1		2
C15: Struggling to identify customer	1						1	1		1	4
C20: Filling Product Owner role								1			1
C17: Struggling with Agile releases			1		1						2

Figure 5.1: Challenges of *Customer Involvement* From the Case Study

The most common challenge was the lack of sufficient *customer involvement* (C14) - identified in eight of the ten case projects. Although the projects tried to implement *customer involvement*, it appeared that it was difficult to acquire a satisfactory level of *customer involvement*, indicating it is a challenging practice to implement. The insufficient *customer involvement* can result from too few customer resources involved in the project (Project 1, 2, 9, and 10), which further cause delays in clarifications (Project 2). The insufficient *customer involvement* can also result from a lack of engagement from the customer or a lack of effort from the project team to engage the customer (Project 3, 5, and 8). It is challenging for the project teams to engage the customer if there are no dedicated customer resources (Project 3) or if they lack understanding of their responsibilities (Project 5 and 8) - hence, connecting the insufficient involvement to the lack of understanding of the practice (C2). Although the customer should take responsibility to involve themselves, the project team has part of this responsibility. Decisions made by the project team impact how difficult it is to give feedback, and the execution of the demos in Project 3 and 5 might not have been the best way to encourage *customer involvement*. The informant from Project 8 admitted that they could have been better at doing user testing early in the project to involve the customer more. Therefore, we argue that both the project team and the customer are responsible for ensuring this engagement. It can be especially challenging to ensure customer engagement when no specific individuals have been appointed as customer resources, as no particular person feels responsible for giving feedback and input. Lastly, insufficient *customer involvement* can be caused by customers lacking time to involve themselves sufficiently (Project 4 and 5). Customer resources that are either part-time involved in the project or external to the project are especially busy. Therefore, they can struggle to designate time for sufficient input to the project teams (Project 4 and 5). The lack of sufficient *customer involvement* was also found in literature (Hoda et al. 2011).

The last common challenges were the lack of clear requirements and needs (C13), the customer's lack of understanding of *customer involvement* (C2), and the struggle to identify the customer (C15) - all identified in four projects.

The lack of clear requirements and needs (C13) can be challenging (Project 2, 4, 5, and 10). The requirements can be poorly defined by the customer (Project 4), due to customers not knowing what they want (Project 5), or due to insufficient *customer involvement* (Project 4). The latter reason creates a link to C14. A large group of customer resources makes it especially challenging for the project team to fully understand the customer's opinion and wishes (Project 5 and 10). We reason that an abundance of customer resources available for the entire project team makes it challenging for the project team to know which customer resources to address for the various matters and makes it difficult for the customer resources to agree on and clearly communicate their requirements and needs. The challenge was also addressed to a great extent in literature (Hoda & Murugesan 2016, Pikkarainen et al. 2008, Tanner & Mackinnon 2015, Drury et al. 2012).

The customer's lack of understanding of *customer involvement* (C2) was confirmed by Hoda & Murugesan (2016) and Hoda et al. (2011). The challenge can be caused by inexperience (Project 3 and 7) and failure to understand the importance of their feedback (Project 3). The customer resources can lack experience with the expected frequency of the feedback (Project 3), specification of requirements (Project 7), or even project methodology (Project 7). When the customer lacks understanding of how to be involved in the project, the result can be insufficient *customer involvement* (C14) - as found in Project 5 and 8.

The project teams' struggle to identify the customer was evident in Project 1, 7, 8, and 10. However, we experienced that finding the answer to "Who is your customer?" was time-consuming in most of the interviews. This indicates that few of the case projects had a clearly defined customer. Surprisingly, the challenge was not identified in the reviewed literature. This might be a result of our project sample. Most of the projects addressing this challenge were public projects (Project 1, 7 and 10) and large-scale (Project 1, 7 and 8) - a project sample that differ from the common project samples in literature.

Consequently, the common challenges of *customer involvement* in Norwegian projects are the following.

- C14: Lacking sufficient customer involvement
- C13: Lacking clear requirements and needs
- C2: Lacking understanding of practice
- C15: Struggling to identify customer

Strategies for the common challenges

Using a customer proxy or a team of customer proxies is a strategy widely described in literature (Cao & Ramesh 2008, Hoda et al. 2011, Layman et al. 2006). Both Project 1 and 7 used customer proxies to ensure customer identification (C15), whereas Project 2 used it to define requirements and needs clearly (C13). Customer proxies can provide the project teams with a clearly defined customer, while also making the customer proxies responsible for identifying the customer's needs and requirements. The strategy also offers a way of reducing a large customer group to a manageable and beneficial number of people. However, the customer proxy team can be too small, and it appears advantageous to establish a customer proxy team with a capacity that matches that of the developer teams to prevent developers from waiting for clarifications (Project 2). Although the customer proxy establishes a clear customer for the project team, we reason that the project team can experience the remaining common challenges with a customer proxy just like with any other customer resource. Hence, implementing a customer proxy gives no guarantee for the successful implementation of *customer involvement*, and it should therefore be combined with other strategies.

A way to cope with unclear requirements and needs (C13) is to give more authority to the project teams (Project 10). Not clarifying all changes with the customer, but trusting the team with some decisions can avoid some of the slow clarifications caused by a lack of customer resources in the project. This implies making the teams more self-organized, indicating the implementation of *self-organized team* as a beneficial strategy.

Having full-time customer resources - e.g., customer proxies or customer representatives - actively involved in the project as part of the project team was suggested as a strategy to ensure sufficient *customer involvement* (C14) in Project 1, 4, and 7. This can again enable clear requirements and needs (C13); By increasing the customer resources' involvement in the project, they have more time to define high-quality requirements and needs (Project 4). Project 1 and 4 had 80 and 10 man-years involved respectively - significantly different sizes - and we can therefore assume that having full-time customer representatives applies to various project sizes. However, acquiring full-time customer resources as active participants in the project can be challenging to achieve, as someone must be willing to invest the required money and time. Perhaps more applicable and less demanding is involving part-time resources in the project team, as proposed by one informant from Project 9. However, this variant of the strategy would not give the same effect as with the full-time resources. Hence, we suggest acquiring full-time resources if this is possible.

Quite contradictory to the proposed active involvement of customer resources in the project team, we have also discovered that it is possible to successfully implement *customer involvement* with external customer resources. Three informants from Project 9 reported this, and the reason for the success was possibly their close and informal communication between the team members and customer. Lastly - and quite interestingly - the active involvement of customer representatives was considered important in Project 1, even though they had an actively involved customer proxy. This indicates that actual customer representatives are more valuable than a customer proxy. We are not surprised, as a customer proxy is in fact an imitation of a customer (Cao et al. 2009, Hoda et al. 2011, Layman et al. 2006). Conclusively, we argue that part-time customer resources involved in the project team is the bare minimum to ensure sufficient *customer involvement* for most projects, and that full-time customer resources (preferably as customer representatives and not customer proxy) is the optimal solution, as long as someone is willing to take the bill.

Increasing the number of customer resources can naturally increase the *customer involvement* (C14). The strategy was suggested in Project 2 and 10. It too requires an investment of time and money, and the strategy is therefore not easily implemented.

Appointing specific individuals as customer resources is a strategy that can be implemented to ensure sufficient involvement of the customer (C14) and identification of the customer (C15). The strategy was proposed for C14 in Project 5 and 10. We found that it is important that specific individuals take responsibility for the various aspects of the project delivery to exploit the potential value-creation (Project 5) and that the appointed individuals are adequate for the role (Project 10). The latter is especially important if only a few are appointed. An adequate customer resource should have long experience in the customer organization and be familiar with the organization's various departments that receive the project delivery, in order to represent the customer satisfactory (Project 10). On the other side, the strategy was suggested in Project 8 and 10 to tackle the lack of identified customer (C15). In Project 8, they assigned people who were considered specialists in their fields as Product Owners to represent the customer satisfactorily. A good Product Owner is someone comfortable involving the various stakeholders and making decisions that leave some stakeholders dissatisfied (Project 8).

Close communication between customer and team members can ensure sufficient *customer involvement* (C14). This is quite contradictory to shielding the team members - a strategy identified by Hoda & Murugesan (2016) - but it aligns well with the frequent in-depth discussion suggested by Tanner & Mackinnon (2015) . Informants from Project 9 claimed that the fact that they knew the customer well, combined with them having a weekly meeting directly with the customer where the team members felt comfortable participating actively, constituted important success factors. Also, using a common digital communication platform - i.e., Slack - was considered to increase the frequency of "smaller" clarifications. This increased communication can compensate for having too few customer resources available in the project. Project 10 also knew their customer well, enabling them to skip intermediaries (customer representatives) when they needed quick clarifications (C14). We consider by-passing the assigned customer representatives a controversial strategy, as one can risk introducing confusion and disorder.

The customer resources should be trained in their roles to ensure a better understanding of their responsibilities (C2). The strategy's usefulness is confirmed by Hoda et al. (2011). This involves making the customer resources aware of the importance of their feedback (Project 5), and training them in how to work in projects (Project 7), how to specify requirements (Project 7), and other matters related to their responsibilities that they are unfamiliar with. The project management can ensure training - e.g., through an introductory course - or the respective team managers can ensure training (Project 7). By increasing the customer's understanding of *customer involvement*, their involvement can increase (C14) - as suggested in Project 5. However, we discovered that it can be challenging to achieve satisfactory training, indicating that it is a complex strategy difficult to implement correctly. We also argue that training the customer resources is easier if the customer resources are part of the project team, as they then answer to the project management like any other team member. It can be more challenging with external customer resources - and perhaps not appropriate - to try to train them.

To make the customer resources understand their responsibilities (C2), performance reviews can be arranged. In Project 5, the customers had to answer for the added value of the project releases, creating awareness of their responsibility to implement the releases and give feedback to the project team.

Task forces can be established to identify who to address for the various matters (Project 7). This strategy is beneficial when the project lacks a clear customer (C15). However, Project 7 had a particularly large number of customer resources, which naturally makes it especially challenging to identify the responsibility areas of each customer resource. Thus, we claim that creating task forces is a valuable strategy for projects with many customer resources, but not very relevant besides that.

Conducting user tests is a way to increase *customer involvement* (C14), as suggested in Project 8. It was considered important to conduct user tests also early in the project.

We thus recommend the strategies presented in Table 5.3 for the common challenges of *customer involvement* in Norwegian projects.

Successful implementation of *customer involvement* in Norwegian projects

Based on the interviews and our own reflections, we recommend the following implementation of *customer involvement* in Norwegian projects:

- Appoint specific individuals as customer resources - e.g., customer representatives, customer proxies, or Product Owners. **Avoid** large and unorganized groups of customer

Challenge	Strategy
C14: Lacking sufficient customer involvement	Having full-time customer resources Increasing the number of customer resources Appointing specific individuals as customer resources Conducting more user tests Having customer resources as part of the project team Establishing close communication between customer and team members
C2: Customer lacking understanding of practice	Training Arranging performance reviews Increasing the authority of project teams
C13: Lacking clear requirements and needs	Using customer proxy Having full-time customer resources Increasing the authority of project teams
C15: Struggling to identify customer	Using customer proxy Establishing task forces Appointing specific individuals as customer resources

Table 5.3: Strategies for *Customer Involvement* From the Case Study

resources, as one can risk that no one really feels responsible for giving feedback and ensure value-creation.

- Include customer resources as part of the project team to ensure the team members and customer can work closely together, thus creating a better understanding of the customer's requirements and needs. **Avoid** having external customer resources, as most projects require closer contact with the customer.
- Involve a sufficient number of customer resources - and preferably full-time - to prevent a bottleneck with team members waiting for clarifications from the customer. However, someone must be willing to invest the required time and money.
- Give more authority to the project teams - e.g., by implementing *self-organized team* - if customer resources are scarce.
- Choose customer resources with care. They must represent the customer well. Therefore, they should have long experience in the customer organization, be familiar with the organization's receiving departments, and be skilled at involving the various stakeholders and reaching decisions that leave some stakeholders dissatisfied.
- Prioritize acquiring customer representatives as the customer resources. The customer representatives can represent the end-users.
- Establish a close and informal relationship with the customer, as this can enable better collaboration.
- Train customer resources in their responsibilities in the project.
- Encourage customer resources to take ownership of the value-creation - e.g., through arranging performance reviews where the customers must answer for the added value.
- Ensure that there are customer resources responsible for the various aspects of the project

delivery. Task forces can identify who to address in projects with a large group of customer resources.

- Ensure that the project team puts enough effort in involving the customer sufficiently - e.g., through conducting user tests early on.

6 Agile Practice: Release Planning

In this chapter, we will target the three research questions defined in Chapter 1 for the selected Agile practice *release planning*, following the research design described in Chapter 2. As we have chosen a nontraditional report structure that we consider beneficial for this study, we will present all related work, results, and discussion of *release planning* in this chapter. To make sure the reader understands the Agile practice under investigation in this chapter, we will commence with a short description of *release planning.*

6.1 Description of Practice

Release planning is a complex task (Heikkilä et al. 2015). It is common for the release plan to describe proposed project releases and a plan for the necessary iterations. However, what is understood as a release, and how *release planning* is implemented in projects, varies greatly. Some projects have releases integrating numerous iterations, whereas other projects release several times a day. Both of these versions, and instances in between, will be studied in this chapter.

To be able to investigate *release planning* adequately across the various project types, we have made some generalizations. Although *release planning* can be implemented in addition to iteration planning (sometimes known as sprint planning), this is not the case for all projects. Many projects release every iteration and do not distinguish *release planning* from iteration planning. Thus, in many projects, *release planning* and iteration planning is the same. Therefore, we have chosen to let *release planning* include iteration planning, hence studying both planning types at the same time. For our study, this means that challenges and strategies connected to iteration planning have been included as challenges and strategies of *release planning.*

6.2 Related Work

In this subchapter, literature related to *release planning* and the research questions will be presented. We have identified the related work by following the methods described in Subchapter 2.2.

6.2.1 Challenges

Challenges of *release planning* have been addressed by Heikkilä et al. (2015), Moe et al. (2012), Drury et al. (2012), Hoda et al. (2011), Ramesh et al. (2010), Grapenthin et al. (2015) and Hoda & Murugesan (2016). These are listed in Table 6.1. The most frequently cited challenges are considered more applicable and robust and will therefore be discussed. The remaining challenges are considered self-explanatory and will not be discussed in further detail.

C7: Lacking clear direction

Difficulties in prioritization of requirements is a common challenge (Heikkilä et al. 2015), as it results in a lack of a clear direction from a strategic level when planning a release. It can lead to the majority of the planning meeting being spent discussing project scope instead of translating the project's strategic goals into technical tasks (Moe et al. 2012). Sometimes, Agile teams are facing opposing requirements from various customers. This complicates the prioritization and release planning further (Drury et al. 2012).

ID	Challenge	References
C1	Dealing with external dependencies	Drury et al. (2012)
C2	Lacking understanding of practice	Heikkilä et al. (2015)
C7	Lacking clear direction	Moe et al. (2012) Heikkilä et al. (2015) Drury et al. (2012)
C14	Lacking sufficient customer involvement	Hoda et al. (2011) Ramesh et al. (2010)
C21	Estimating tasks	Grapenthin et al. (2015) Hoda & Murugesan (2016) Ramesh et al. (2010)
C22	Lacking alignment with strategic plan	Moe et al. (2012) Ramesh et al. (2010)
C23	Obtaining realistic backlog	Moe et al. (2012)
C24	Missing Definition of Done	Moe et al. (2012)
C25	Not committing to plan	Moe et al. (2012)
C26	Lacking sufficient architectural planning	Heikkilä et al. (2015)
C27	Having inefficient release planning meetings	Heikkilä et al. (2015)

Table 6.1: Challenges of *Release Planning* From Literature

C14: Lacking sufficient customer involvement

Release planning includes writing and prioritizing user stories, which depends on having an involved customer in the project Hoda et al. (2011). For more information on lack of *customer involvement* and optimizing it (see Chapter)5.

C21: Estimating tasks

Task estimation can be challenging (Hoda & Murugesan 2016). When combining *release planning* with *self-organized team* - i.e., executing collective estimation - the team manager can struggle to align the different estimations. Another challenge is that task estimations can be time-consuming (Hoda & Murugesan 2016), thus stealing time that could have been spent on other activities that possibly can create more value. A third challenge is that estimates can often simply be wrong or inaccurate (Grapenthin et al. 2015, Ramesh et al. 2010). Grapenthin et al. (2015) also observed that about a quarter of all tasks within an iteration were identified after the iteration was initiated, which led to further deviations from the time estimates.

C22: Lacking alignment with strategic plan

Aligning strategic and operational plans can be difficult in practice (Moe et al. 2012). First, it relies on both the business and technical departments understanding the delivery. Sometimes the planning meeting is not enough for everyone to understand the complexity of the delivery. This might again lead to confusion and plans changing frequently (Moe et al. 2012). Second, code quality is sometimes sacrificed for visible progression (Moe et al. 2012, Ramesh et al. 2010). This often happens when there is no arena for solving conflict between strategic and operational levels, leading to the strategic progress plan overruling. Third, many organizations are stuck in traditional, hierarchical decision processes and do not involve the technical team when making strategic decisions. This can lead to delays, and in the worst case, unusable deliveries (Moe et al. 2012).

6.2.2 Strategies

Strategies for *release planning* have been addressed by Masood et al. (2020), Grapenthin et al. (2015), Moe et al. (2012), Golfarelli et al. (2013) and Ramesh et al. (2010). These are listed in Table 6.2. The IDs in parentheses show what challenges each strategy deals with, and those without parenthesis are enablers of the practice in general and do not target any specific challenges. The most frequently cited strategies are considered more applicable and robust and will therefore be discussed. This applies to none of the strategies. All strategies are therefore considered self-explanatory and will not be discussed in further detail.

Strategy	References
Having enough tasks available in each iteration (C22)	Masood et al. (2020)
Sufficient detailing of backlog items (C21)	Grapenthin et al. (2015)
Involving all stakeholders in planning meetings (C21)	Grapenthin et al. (2015)
Ensuring stakeholders' understanding of domain (C21)	Moe et al. (2012)
Using digital optimization models	Golfarelli et al. (2013)
Strengthening shared mental model (C22)	Moe et al. (2012)
Delaying strategic decisions (C23)	Moe et al. (2012)
Adopting change management policies	Ramesh et al. (2010)

Table 6.2: Strategies for *Release Planning* From Literature

6.3 Results

In this subchapter, the results from the case study related to *release planning* will be presented and discussed. We will identify challenges and associated strategies, and the IDs in parentheses behind the strategies show what challenges they deal with. The results from each project will be addressed separately.

Project 1

Release planning meetings are dependant on feedback from team members. Nevertheless, team members in Project 1 often stayed passive during the meetings. For one of the informants, the threshold for voicing an opinion during the *release planning* meetings was high. The reason was the feeling of lack of necessary competence to offer good feedback.

It can be challenging to have several releases in parallel. The project could work on as many as 40 releases at the same time. Two informants reported that it was challenging to coordinate all the releases they had in parallel. Two different software solutions had to be used to maintain an overview of all the releases in progress. The complex coordination led to a lack of control of both what each release consisted of and when to release.

Estimating tasks consistently across teams was a challenge. The project estimated tasks in story points, based on what elements the tasks consisted of. Examples of such elements were technical complexity and how many user groups would be affected by the features. An informant considered it challenging that the estimations seldom gave a representative picture of the workload. This led to unrepresentative progress reports and conflicts between team managers due to differing estimations. Furthermore, it decreased motivation because little kudos was given for completed tasks of few story points. As a mitigation strategy, the project tried to specify the estimation guidelines further. Yet, they never reached a good agreement. I9 contemplated estimating in hours instead of story points, without being able to conclude which option was the best:

> It has been hard to find an unambiguous way to estimate improvement because all improvements are different. Some want to estimate in calendar time, others in story points. Still, I think the estimation was good enough for its purpose, even though it wasn't 100% unambiguous and identical. (I9, P1)

Poor estimations made the backlog less realistic. However, the project had a low threshold for moving unfinished tasks into the next iteration. This was considered a satisfactory solution to the problem.

The release planning in Project 1 was further complicated because of dependencies to customer. Their customers - executive officers working in operation - were not ready for the releases and were not able to receive them satisfactorily, partly due to an internal reorganization. As a result, they struggled to provide the project team with necessary feedback for future releases. Lacking feedback from customer became a bottleneck for the project. Consequently, release plans had to be changed mid-release. Close communication with the customer, including introductory courses on the new releases, was used as a strategy. It was considered the Project Management's responsibility, not the developer teams, to prepare the customer for releases.

Conclusively, the following challenges were addressed in interviews with Project 1:

- C8: Having passive team members
- C28: Controlling releases in parallel
- C21: Estimating tasks
- C23: Obtaining realistic backlog
- C1: Dealing with external dependencies
- C14: Lacking sufficient customer involvement (see Chapter 5)

The following strategies were suggested in interviews with Project 1:

- Utilizing tools (C28)
- Establishing guidelines for estimation (C21)
- Estimating tasks in hours (C21)
- Moving unfinished tasks to next iteration (C23)
- Training (C14)

Project 2

Project 2 experienced challenges related to external dependencies. The customers wanted a big, traditional code deployment. I2 shared how this caused several difficulties for the project:

> We struggled a bit with our big deployment. There was a lot of administrative work around it. And a lot to keep an overview of several hundred tasks that are supposed to be of a certain quality. And it's a lot of work in a short amount of time. And it becomes a huge job test-wise, because we have to run safety tests of quality tests, user tests. [...] So you can't test a big solution in three days. Weeks are spent, weeks in calendar time. They [the customers] were very comfortable with working in this way. They felt like they had control, but you really don't have control because it is too big. And the solution, as I mentioned, had to be put together with elastics and paper clips, which we had to fix afterwards. (I2, P2)

As I2 describes, the customers wanted to have one big release. Unfortunately, this lead to a lack of control, the necessity to re-do solutions, and a great administrative workload. Later in the project - after the big deployment discussed above - the project team convinced the customer to plan smaller releases. After the customer got accustomed to smaller deliveries, they recognized the benefits it gave them.

Project 2 utilized a customer proxy team to define requirements. Delayed product requirements from the customer proxy team made it challenging to fill up an iteration plan with tasks. Due to a lack of resources in the customer proxy team, tasks were not defined in time for the iteration planning meetings. To avoid idling, the team performed maintenance tasks. Nevertheless, progression on project tasks stopped. Another suggested strategy was to train the developers in the business fields they were delivering to. In this way, they would be able to specify some of the tasks without waiting for the customer proxy team.

Lack of clear prioritization was also challenging. For example, a maintenance task would be ordered at the same time, and on the same system, as a functional upgrade. The developer team would have to choose which of the two tasks to prioritize, with both the technical perspective and the stakeholder expectations in mind. The team solved this challenge case-by-case by prioritizing what they thought were the most urgent tasks. The Product Owners were also expected to present their view on the prioritization during the planning meetings.

Conclusively, the following challenges were addressed in interviews with Project 2:

- C1: Dealing with external dependencies
- C14: Lacking sufficient customer involvement (see Chapter 5)
- C7: Lacking clear direction

The following strategies were suggested in interviews with Project 2:

- Executing maintenance tasks while waiting (C14)
- Training (C14)
- Having a skilled Product Owner (C7)

Project 3

All team members had responsibilities outside of the project. Often, they prioritized these tasks over the project, and the time they dedicated to the project varied. This made making a realistic backlog challenging.

Furthermore, the project struggled with lacking input on prioritization. The customer did not offer much feedback, which left the project ”guessing” what would be important for the customer.

Conclusively, the following challenges were addressed in interviews with Project 3:

- C1: Dealing with external dependencies
- C23: Obtaining realistic backlog
- C14: Lacking sufficient customer involvement (see Chapter 5)

No strategies were suggested in interviews with Project 3.

Project 4

The team's Product Owner lacked decisiveness, and this led to confusion regarding the prioritization of tasks. Participants in the *release planning* meetings were also often passive. Choosing a confident and engaging meeting leader was suggested as a strategy.

Furthermore, Project 4 was affected by many regulations and had many stakeholders. Consequently, they often had to deal with external dependencies. An important strategy in their project was making a separate list of "development tasks". These were tasks that needed external clarifications. The Product Owner and team manager underlined the importance of starting these tasks early in the project to the developers - much earlier than what the developers thought necessary. This strategy mitigated the consequences of slow clarifications from external partners.

I5 expressed scepticism towards estimation:

> We [team] decided initially that we would not estimate, because we think it is a waste of time. It's just guessing, and it doesn't really mean anything. People aren't able to follow the estimates anyway. We don't want to waste our time on it, because it takes a lot of time. (I5, P4)

The team also skipped reporting time spent on each task. The result was limited details in the progress reports. It was indicated that Project Management preferred more extensive data on estimated and spent time in the various stages of development. The team, however, was very fond of omitting estimation and time tracking of tasks completely.

Conclusively, the following challenges were addressed in interviews with Project 4:

- C7: Lacking clear direction
- C8: Having passive team members
- C1: Dealing with external dependencies
- C21: Estimating tasks

The following strategies were suggested in interviews with Project 4:

- Having a skilled facilitator (C8)
- Highlighting task dependencies (C1)
- Omitting task estimation (C21)

Project 5

The user stories lacked the necessary details. It was the developers who wrote the user stories, and they were not particularly good at it. Two reasons for this were identified; The developers had received little training, and they were not used to focusing on customer value. Organizing various courses was used as a strategy. Yet, attendance at these courses was low. Another strategy was convincing team members of how user stories could aid them in translating feedback from customer into prioritized tasks, as explained by I7:

> But what I'm trying to do, is to get them to understand that this [user stories] is a tool that can help them. That it isn't like something they should do because I say so, but that I genuinely want to help them to translate the feedback they get from

> customer into real user stories which they can prioritize. Rather than just starting working on different things completely without thought. (I7, P5)

Some of the teams in Project 5 estimated tasks, and they estimated in story points. The quality of the estimations was considered poor, and various factors explained this unsuccessful estimation. First, the team members lacked faith in the usefulness of the estimations. Second, most of the tasks demanded contributions from third parties, thus complicating the estimations. An informant wondered whether the execution of task estimation was in fact not appropriate in their project context.

The customer was not used to receiving unfinished products and offering feedback, affecting the *release planning*. This challenge in Project 5 is explained in further detail related to *customer involvement* in Chapter 5.

It was challenging to obtain a realistic backlog. Often, the teams would be too optimistic and plan to work on too many user stories in each iteration. An informant estimated that about 30% of the user stories remained unfinished at the end of each iteration. Defining user stories with a limited scope was suggested as a strategy.

Conclusively, the following challenges were addressed in interviews with Project 5:

- C2: Lacking understanding of practice
- C21: Estimating tasks
- C17: Struggling with Agile releases (see Chapter 5)
- C23: Obtaining realistic backlog

The following strategies were suggested in interviews with Project 5:

- Training (C2)
- Omitting task estimation (C21)
- Limiting the scope of user stories (C23)

Project 6

Some products can be challenging to release in iterations. Project 6 struggled with this issue, as they delivered a product consisting of both software and hardware. For the team members working on the hardware, planning releases and working in short iterations made little sense. Therefore, it was decided to move somewhat away from an iterative methodology, reducing the focus on finishing tasks within an iteration. I8 contemplated whether it was a good idea or not:

> So that's maybe a weak point; there are no formalities. There is not much focus on tasks being finished within an iteration. Personally, I don't know if it's a big problem. I have worked in projects where it was a very, very tight time pressure, and "iteration" was a very uncomfortable word at times. (I8, P6)

Moving away from the iterative focus was viewed as unsatisfactory from the Project Management perspective, as it implied less control.

Finally, the *release planning* meetings were sometimes inefficient. A suggested strategy was to "groom" the backlog before the meeting. Having an opinion of what to be released in the following iteration, and using this to control the discussions in the *release planning* meetings, was considered to possibly increase the meeting's efficiency.

Conclusively, the following challenges were addressed in interviews with Project 6:

- C1: Dealing with external dependencies
- C27: Having inefficient release planning meetings

The following strategies were suggested in interviews with Project 6:

- Executing backlog grooming before release planning meetings (C27)

Project 7

"Scope creep" - unplanned increases in project scope - posed challenges. Sometimes, the scope of the releases would grow due to input from external stakeholders. In these situations, there was not a lot the team could do besides downgrading other tasks. Other times, the release scope grew due to team members adding tasks that were not strictly necessary. To avoid scope creep, it was decided to focus heavily on the usefulness of different features against the overall project goals. Always asking *"Do you need this to deliver the product at the deadline? Yes or no?"*, was a helpful strategy. Yet, having a Product Owner who was good at prioritizing was considered the most important strategy.

Estimation uncertainty was a challenge. The estimation was less challenging as the project matured and gained more experience. Using the "project tolerance" philosophy from PRINCE2 was helpful. This strategy gave the project room to maneuver within the scope and define "must-haves" and "nice-to-haves". Furthermore, the strategy helped them manage sudden changes in scope, as it made it easier to remove tasks that were not "must-haves".

Conclusively, the following challenges were addressed in interviews with Project 7:

- C23: Obtaining realistic backlog
- C21: Estimating tasks

The following strategies were suggested in interviews with Project 7:

- Focusing on strictly necessary features (C23)
- Having a skilled Product Owner (C23)
- Including tolerances in estimates (C21)

Project 8

Project 8 struggled with a highly risk-focused culture. Consequently, the organization was not used to iterative releases. I11 described how they struggled to release features as often as they had planned:

> So, what happened was that instead of releasing after every iteration, some chose to wait, like to reduce the ... I won't call it the administration, but all the work that had to be done to move new functionality to production. So we ended up not releasing anything in months. So that's something we [the management] weren't very conscious about, and then the teams got to choose themselves, and then the Product Owners chose to just postpone and postpone these releases. (I11, P8)

Moving features between environments - i.e., from development, to testing, and later to production - was thus time-consuming. This prevented the Product Owners from releasing frequently. The inability to release frequently worried the team, as the risk of the release being imperfect grew. A suggested strategy to tackle this challenge was to make project resources experienced with Agile frameworks responsible for ensuring frequent releases. Project Management also began demanding frequent releases, while also giving the teams more time to prepare for releases by reducing the amount of new functionality in each release.

Conclusively, the following challenges were addressed in interviews with Project 8:

- C1: Dealing with external dependencies

The following strategies were suggested in interviews with Project 8:

- Dedicating time to release (C1)
- Defining expected release frequency (C1)

Project 9

Project 9 was limited by external dependencies related to the customer and to the project culture. It was important for the project's customer to know what features they would get for their money. Therefore, all features were planned a year in advance. Consequently, the team had little room to change features or the release plan according to new information that appeared. Integrating the Agile project with the customer's waterfall approach was perceived as a great challenge. Regarding the project culture, Project 9 started out with a culture that valued finishing as many tasks from the backlog as fast as possible. Unfortunately, this caused many bugs, which had to be dealt with afterwards. To decrease the number of generated bugs, they started pre-releasing to the customer when the releases were 90% finished. Consequently, the customer was involved earlier in user tests, leaving the team with more time to fix the bugs before the actual release. Moreover, they started focusing on not running a risk late in an iteration; If a release did not meet the 90% quality-criteria, it would be released in the next iteration instead. The strategy resulted in adjustments of the work culture.

We found that some team members could be too passive in the *release planning* meetings. The team manager would try to involve all team members. Sometimes, team members were assigned to be in charge of task definition. This increased the participation slightly. Having an engaging meeting facilitator that gathered opinions from all participants was perceived as important.

Inaccurate estimations constituted a challenge, as reported by two of the informants. The rough estimates executed before the project initiation were too low, and this was discovered when the tasks were further specified in the *release planning*. Consequently, the team had to consult the customer to evaluate whether scope changes would be necessary. The challenge could be mitigated by over-estimating tasks and with the team gaining experience.

The team members of Project 9 worked on several projects in parallel. Consequently, prioritization was a challenge. When urgent matters emerged in the case project, it was unclear if these were to be prioritized over tasks from other projects. I17 explained this:

> It's always a challenge that the customer wants things that arrive late. Yes. And a little bit related to the capacity of other projects, like "What should we prioritize? Should we prioritize [case project], or do we have to prioritize other projects?" So the prioritization is always a challenge. (I17, P9)

Furthermore, urgent tasks increased the technical dept of other tasks. This challenge was also improved by over-estimating and thus having room for urgent tasks.

Conclusively, the following challenges were addressed in interviews with Project 9:

- C1: Dealing with external dependencies
- C8: Having passive team members
- C21 Estimating tasks
- C23: Obtaining realistic backlog
- C7: Lacking clear direction
- C29: Having technical debt

The following strategies were suggested in interviews with Project 9:

- Involving customer before finishing release (C1)
- Giving team members specific responsibilities (C8)
- Having a skilled facilitator (C8)
- Over-estimating (C21, C29)

Project 10

Lack of *customer involvement* created a bottleneck in the project. The project defined user stories and acceptance criteria together with their customers. Therefore, the lack of *customer involvement* caused a lack of user stories ready for development, which several informants addressed. The root cause was the late start of the project, as shared by I21:

> Because we started the project a little late, and as a consequence, we have always been a bit behind on having things ready for the next sprint. And it's especially those resources that are doing the preparations of specifying "this is how the screenshots should look, this is how the functional things should work, this is how it's broken down into actual tasks that can actually be developed, and this is how long it will take". (I21, P10)

Consequently, developers would end up having to wait for new tasks. Introducing a formalized agreement of *customer involvement* improved their situation. Another informant explained that the most important strategy was allocating fewer project resources to feature development and more resources to planning and specification with the customer. Moreover, the user stories were of low quality, possibly due to the lack of a Product Owner. At the time of the interviews, the Product Owner's responsibilities were distributed between various team members. Hiring a consultant to take the role of Product Owner was suggested.

Project 10 commenced their project with technical debt because the features they developed were based on an already existing system with unfixed bugs. I20 described this challenge:

> The biggest problem is that we have things in operation before we start the project, that still aren't done, where one finds bugs all the time that delays the deployment of the release. [...] In other words, it's old issues that linger. (I20, P10)

Consequently, deliveries were delayed, and the release plan was disorganized. The team would then also struggle to have an overview of unfixed bugs and of what versions were being sent to acceptance testing. If given the time, they wished to have addressed all the technical debt of the existing system before starting the project.

Conclusively, the following challenges were addressed in interviews with Project 10:

- C14: Lacking sufficient customer involvement (see Chapter 5)
- C29: Having technical debt

The following strategies were suggested in interviews with Project 10:

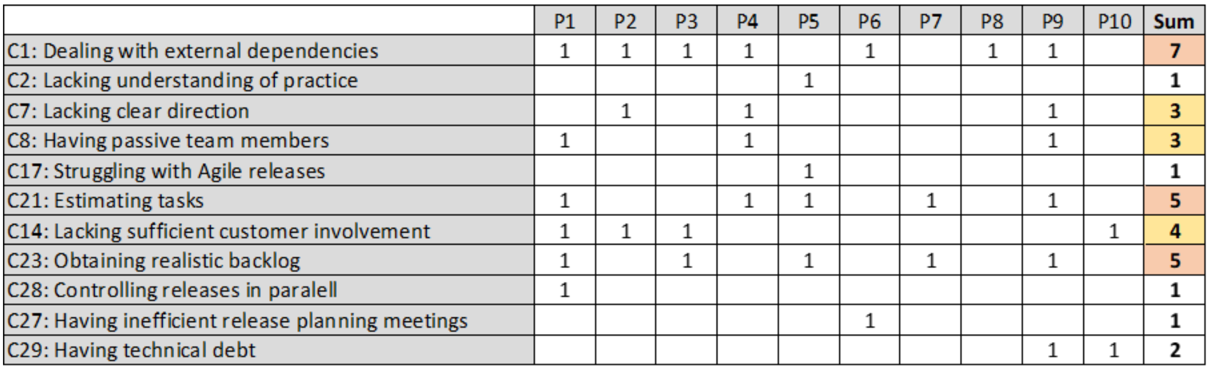
- Formalizing agreement of customer involvement (C14)
- Allocating sufficient resources for task specification (C14)
- Having a skilled Product Owner (C14)
- Resolving existing technical debt before project initiation (C29)

6.4 Discussion

In this subchapter, we will conduct a cross-case analysis where we discuss our findings and connect them to the identified related work. We will also answer our three research questions with respect to *release planning.*

Common challenges of *release planning* in Norwegian projects

We identified various challenges in interviews with the ten case projects. Figure 6.1 shows the number of projects where the challenges have been identified. Challenges found in three or more projects are considered common. The sum is yellow for challenges identified in 3-4 projects and red for those identified in five or more. The remaining challenges - with white sums - are considered uncommon. We will only discuss the common challenges.



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Sum
C1: Dealing with external dependencies	1	1	1	1		1		1	1		7
C2: Lacking understanding of practice					1						1
C7: Lacking clear direction		1		1					1		3
C8: Having passive team members	1			1					1		3
C17: Struggling with Agile releases					1						1
C21: Estimating tasks	1			1	1		1		1		5
C14: Lacking sufficient customer involvement	1	1	1							1	4
C23: Obtaining realistic backlog	1		1		1		1		1		5
C28: Controlling releases in paralell	1										1
C27: Having inefficient release planning meetings						1					1
C29: Having technical debt									1	1	2

Figure 6.1: Challenges of *Release Planning* From the Case Study

The most common challenge was dealing with external dependencies (C1). We identified the challenge in seven of the ten case projects. Various dependencies can affect the *release planning* negatively. First, the *release planning* can suffer from dependencies to customer (Project 1, 2, and 9). The customer can struggle to give the necessary feedback that the development teams need for *release planning*, causing changes mid-release (Project 1). The customer can also wish for a more Waterfall-approach with big and seldom releases, making it challenging for the teams to release in an Agile way (Project 2 and 9). Project 9 struggled with a customer insisting on waterfall planning, which prevented the team from adjusting release plans underway in the project, even if it was to increase the customer value of the product. Second, the culture can hinder the successful implementation of *release planning* (Project 8 and 9). Project 8 struggled with an organizational culture working against the *release planning.* Their organization was not used to or comfortable with releasing in small iterations, and the teams would try to avoid releasing small and "unfinished" deliveries. Consequently, the project ended up having big and *"scary"* releases. Project 9 had a somewhat opposite cultural problem; Their organization prioritized quantity over quality, and they would release too fast and with numerous bugs. Third, project resources working with obligations outside the project too can be a barrier to *release planning* (Project 3). As the project resources in Project 3 only worked part-time in the case project, the availability of the resources was unpredictable, causing delays in the release plans. This prioritization of externals tasks over project tasks is confirmed by Drury et al. (2012). Some products can be challenging to release in an Agile way (Project 6). The short iterations did not suit the hardware development of Project 6 well, making it challenging to plan in releases for the hardware parts of the project product. Furthermore, having many dependencies poses challenges for the *release planning* (Project 4). Delivering a product affected by many regulations and demanding many stakeholders' involvement add uncertainties to the release plan.

The challenges of estimating tasks (C21) and obtaining a realistic backlog (C23) were both identified in five projects.

Estimation (C21) is time-consuming, as found in Project 4, literature, (Hoda & Murugesan 2016). Thus, project participants sometimes struggle to understand why it is necessary (Project 5), especially as it seems quite common to have inaccurate estimations which make the backlog unrealistic (C23). Inaccurate estimations have been addressed both in literature (Grapenthin et al. 2015, Ramesh et al. 2010) and our multiple-case study (Project 5, 7, and 9). For example, too optimistic estimations cause unrealistically long iteration backlogs (Project 5). Furthermore, tasks are sometimes dependant on deliveries from external stakeholders, and estimating correctly on their behalf is difficult (Project 5). It is also difficult to align estimation methods within a team (Hoda & Murugesan 2016) and across teams (Project 1). Lastly, it is common that new tasks emerge mid-iteration, creating the need to adjust the backlog (Grapenthin et al. 2015).

The challenge of obtaining a realistic backlog (C23) is confirmed by literature (Moe et al. 2012). In our multiple-case study, we found that the challenge often emerges as a result of other challenges. Poor task estimation (C21) causes the backlog to become unrealistic (Project 1). The same can happen due to external dependencies (C1), as there is a negative impact from project resources having to prioritize external tasks over the project's tasks (Project 3). In Project 5, the team had a hard time including an appropriate number of user stories, resulting in a too optimistic backlog. Furthermore, scope creep makes the backlog bigger than first intended (Project 7).

Lacking sufficient *customer involvement* (C14) was also identified as a common challenge of *release planning* in four of the projects. Customers are needed to write good user stories and prioritize efficiently (Hoda et al. 2011), thus making sufficient *customer involvement* important to plan releases correctly and efficiently. Lacking feedback from customer causes difficulties in prioritizing (Project 3). Delayed feedback from customer (Project 1) and delayed requirement specification from customer (Project 2 and 10) also constitute bottlenecks that lead to re-planning.

The last common challenges were the lack of clear direction (C7) and passive team members (C8).

Similar to the lack of *customer involvement* (C14), the lack of clear direction (C7) makes it difficult to prioritize (Heikkilä et al. 2015). The lack of clear direction can result from Conflicting preferences of various stakeholders (Project 2) - confirmed by Drury et al. (2012) - or absence of direction from organizational management for teams that work on several projects in parallel

(Project 9). Factors within the team can also cause the lack of direction; Project 4 experienced having an indecisive Product Owner that negatively impacted the team's sense of direction.

Some team members do not actively participate in the *release planning* meetings (C8). We identified this challenge in Project 1, 4, and 9. Team managers might wish to get feedback from all participants (Project 9), but some believe they do not possess the necessary competence and experience to offer high-quality feedback (Project 1).

Consequently, the common challenges of *release planning* in Norwegian projects are the following:

- C1: Dealing with external dependencies
- C21: Estimating tasks
- C23: Obtaining realistic backlog
- C14: Lacking sufficient customer involvement
- C7: Lacking clear direction
- C8: Having passive team members

Strategies for the common challenges

A variety of strategies emerged in the interviews, and most of them were only mentioned once. Instead of dedicating a paragraph to each of the identified strategies - as done in the remaining practice-chapters - we will present all strategies for each common challenge in separate paragraphs.

Several strategies were suggested to deal with the external dependencies (C1). A strategy to cope with tasks with external dependencies is to highlight these dependencies (Project 4). That way, these tasks can be started early, as getting the necessary clarifications can be time-consuming. We also found this importance of highlighting task dependencies in literature concerning *self-organized team* (Masood et al. 2020, Hoda & Murugesan 2016). In Project 8, they reduced the amount of new functionality of the releases, as releasing can be a time-consuming process. Thus, more time was dedicated to releasing. For teams unfamiliar and uncomfortable with Agile releases, it is especially important to dedicate enough time to release and to clearly communicate release frequency expectations (Project 8). However, we argue that these two strategies are self-explanatory for projects with resources that have more experience with Agile releases. In Project 9, they started involving the customer before the releases were completely finished, as a way to increase the quality focus.

Estimation continues to be a complicated activity (C21), and we were unable to identify strategies from the case projects that dramatically improved their estimations. Guidelines for estimation can be established, to ensure a common understanding of how to estimate across teams. However, it is difficult to agree on such guidelines, as seen in Project 1. We reason that these guidelines should have been established prior to project initiation and not after conflicts between team managers had arisen. We also believe that it can be especially challenging to establish common guidelines when estimating in story points, as story points can be less quantifiable and more abstract than a time-unit for many. However, this does not mean that we believe story points should not be used. Our results show no clear correlation between used estimation unit and estimation challenges. We find it difficult to say if estimation is best done in story points or in a time unit. Therefore, we do not recommend one estimation unit over the other, hence discarding the strategy of estimating in hours (Project 1). As teams can perceive estimation as challenging, time-consuming, and of little value, several projects suggested omitting this activity completely (Project 4 and 5). This strategy is most likely only suitable in some project contexts, as it poses new challenges for Project Management, whose control is reduced. To cope with inaccurate estimation, a strategy can be to include tolerances in the estimates - e.g., according to PRINCE2 (Project 7). Another strategy is to over-estimate tasks, as a way to increase the team members' flexibility while avoiding delays for the customer (Project 9). Conclusively, we have identified some strategies that address the estimation challenge in various ways. Still, we do not claim that they make the estimates more accurate, nor mitigate the challenge completely, as there will always be uncertainties in estimates. However, they can aid in reducing the negative impact from the challenge.

We have identified several strategies for unrealistic backlogs (C23). Having a low threshold for moving tasks to the next iteration is a way to increase the flexibility of the backlog (Project 1). Limiting the scope of user stories can make it easier to know how many user stories will be completed within an iteration (Project 5). Avoiding scope creep by focusing on doing strictly necessary tasks, and having a Product Owner who can prioritize and enforce this, can also help make the backlog more realistic (Project 7). The use of tolerance We wish to underline that all these strategies can only make the backlogs *more* realistic.

We have identified that the lack of customer involvement (C14) affects *release planning* negatively. The vast majority of findings on how to increase *customer involvement* can be found in Chapter 5. In this chapter, we have identified some strategies to increase *customer involvement* with regard to *release planning.* As they do not give the full picture of how to deal with insufficient *customer involvement*, we will not discuss these further, but rather refer the reader back to Chapter 5.

We only identified one strategy that directly addressed the lack of clear direction (C7); Having a decisive and clear Product Owner taking responsibility for prioritization can make the situation clearer for team members (Project 4). For passive team members (C8), a skilled meeting leader can be beneficial (Project 4 and 9). The facilitator should be confident and engaging (Project 4). Giving team members specific responsibilities in the meetings can enable increased participation, although it was seen to only work slightly in Project 9. Furthermore, the high threshold to speak in the *release planning* meetings in Project 1 indicate that building a trustful team culture can make team members less passive.

Successful implementation of *release planning* in Norwegian projects

Based on the interviews and our own reflections, we recommend the following implementation of *release planning* in Norwegian projects:

- Highlight task dependencies, and start earlier than expected on these tasks, to give time for necessary clarifications.
- Dedicate sufficient time to release, as releasing is a time-consuming process. **Avoid** introducing too much new functionality in each release.
- Clearly define release frequency to project teams and to customer, and what this release frequency entails for them.
- Consider involving customer before the release is completely finished, to get valuable feedback and increase the quality of the release.

Challenge	Strategy
C1: Dealing with external dependencies	Highlighting task dependencies Dedicating time to release Defining expected release frequency Involving customer before finishing release
C21: Estimating tasks	Establishing guidelines for estimation Omitting task estimation Including tolerances in estimates Over-estimating
C23: Obtaining realistic backlog	Moving unfinished tasks to next iteration Limiting the scope of user stories Focusing on strictly necessary features Having a skilled Product Owner
C14: Lacking sufficient customer involvement	*See Chapter 5*
C7: Lacking clear direction	Having a skilled Product Owner
C8: Having passive team members	Having a skilled facilitator Giving team members specific responsibilities Building appropriate culture

Table 6.3: Strategies for *Release Planning* From the Case Study

- Assess the value of estimation versus the development team's required effort. Perhaps the team's effort is greater than the value for project management and the various stakeholders. Consider therefore whether omitting estimation is of greater total value.

- If choosing to estimate; Establish guidelines for estimation prior to project initiation, and plan for the uncertainty of task estimations - e.g., by including tolerances and aiming to over-estimate instead of under-estimate.

- Make the backlog more realistic by limiting the scope of user stories and focusing on strictly necessary features. Accept that some tasks might be moved to the next iteration if the backlog is too optimistic.

- Invest in important roles, such as Product Owners and the facilitators of the *release planning* meetings. The Product Owner should provide the team with a clear direction, and the meeting facilitator should be confident and engaging.

- Build a culture that makes everyone comfortable participating actively in the *release planning*, and give team members specific responsibilities in the *release planning* meetings.

7 Agile Practice: Retrospectives

This chapter will target the three research questions defined in Chapter 1 for the selected Agile practice *retrospectives*, following the research design described in Chapter 2. As we have chosen a nontraditional report structure that we consider to be beneficial for this study, we will present all related work, results, and discussion of *retrospectives* in this chapter. To make sure the reader understands the Agile practice under investigation in this chapter, we will commence with a short description of *retrospectives*.

7.1 Description of Practice

Retrospectives are the meetings Agile teams hold to reflect on, and thereby improve, processes and issues within the team (Jovanović et al. 2016). The aim of *retrospectives* is to identify measures to ensure this improvement. In Scrum, *retrospectives* are arranged at the end of each iteration, with the team discussing what went well, what could be improved, and what corrective actions to commit to (Scrum.org 2021). However, as we will describe in this chapter, some choose other frequencies for their *retrospectives*.

7.2 Related Work

In this subchapter, we will present literature related to *retrospectives* and the research questions. We have identified the related work by following the methods described in Subchapter 2.2.

7.2.1 Challenges

Challenges of *retrospectives* have been addressed by Lehtinen et al. (2017), Bjarnason et al. (2014) and Drury et al. (2012). These are listed in Table 7.1. The most frequently cited challenges are considered more applicable and robust and will therefore be discussed. The remaining challenges are considered self-explanatory and will not be discussed in further detail.

ID	Challenge	References
C30	Dealing with problems outside the team's control	Lehtinen et al. (2017)
C31	Identifying root causes	Lehtinen et al. (2017)
C32	Lacking appropriate corrective actions	Lehtinen et al. (2017) Drury et al. (2012)
C33	Reaching long-term decisions	Bjarnason et al. (2014)
C34	Having recurring discussions	Lehtinen et al. (2017)
C35	Lacking follow-up of corrective actions	Drury et al. (2012)
C36	Having biased discussions	Lehtinen et al. (2017)

Table 7.1: Challenges of *Retrospectives* From Literature

C32: Lacking appropriate corrective actions

It is challenging to identify appropriate corrective actions in *retrospectives* (Lehtinen et al. 2017). Sometimes, few corrective actions are identified, even though many problems are discussed (Lehtinen et al. 2017). This can happen if the *retrospectives* are used more for venting frustration than reaching decisions (Drury et al. 2012). Other times, the identified corrective actions do not address the problems they are meant to tackle (Lehtinen et al. 2017). Corrective actions can also deal with the visible symptoms of a problem instead of its actual cause, hence identifying corrective actions without discussing if they in fact address the root cause of the problem (Lehtinen et al. 2017). Thus, the inability to identify root causes (C31) results in inappropriate corrective actions. Lehtinen et al. (2017) also observed that more corrective actions were identified for the problems perceived as controllable for the team, indicating the challenge of uncontrollable problems (C30) as a root cause of the lack of corrective actions.

7.2.2 Strategies

Strategies for *retrospectives* have been addressed by Lehtinen et al. (2017), Bjarnason et al. (2014), Jovanović et al. (2016), Gupta et al. (2019), Babb et al. (2014), and Drury et al. (2012). These are listed in Table 7.2. The IDs in parentheses show what challenges each strategy deals with, and those without parenthesis are enablers of the practice in general and do not target any specific challenges. The most frequently cited strategies are considered more applicable and robust and will therefore be discussed. The remaining strategies are considered self-explanatory and will not be discussed in further detail.

Strategy	References
Building appropriate culture	Gupta et al. (2019)
Having a skilled facilitator	Bjarnason et al. (2014) Jovanović et al. (2016)
Basing discussions on factual evidence (C36)	Bjarnason et al. (2014) Lehtinen et al. (2017)
Reflecting on biases potentially affecting discussions (C36)	Lehtinen et al. (2017)
Focusing on controllable problems (C30)	Lehtinen et al. (2017)
Collaborating with relevant people outside team (C30)	Lehtinen et al. (2017) Bjarnason et al. (2014)
Introducing games	Jovanović et al. (2016)
Identifying corrective actions for root causes (C34)	Lehtinen et al. (2017)
Identifying types of recurring discussions (C34)	Lehtinen et al. (2017)
Introducing long-term perspective (C33)	Bjarnason et al. (2014)
Adjusting meeting site to format	Lehtinen et al. (2017)
Keeping retrospectives short	Lehtinen et al. (2017)
Microblogging on wiki page	Babb et al. (2014)
Creating clear corrective actions with owners (C35)	Drury et al. (2012)

Table 7.2: Strategies for *Retrospectives* From Literature

Having a skilled facilitator

It is important to have a skilled facilitator of the *retrospectives* (Jovanović et al. 2016, Bjarnason et al. 2014). The facilitator should create an environment that enables a valuable outcome (Jovanović et al. 2016) by being knowledgeable about the topics discussed (Bjarnason et al. 2014). Furthermore, the facilitator should adjust the focus of the discussions to the current needs of the team (Jovanović et al. 2016). Jovanović et al. (2016) proposed having a person outside the team as the facilitator to prevent unnecessary discussions.

Basing discussions on factual evidence (C36)

A strategy for the challenge of biased discussions (C36) is to base the discussions on factual evidence - e.g., evidence-based timelines (Lehtinen et al. 2017, Bjarnason et al. 2014). An evidence-based timeline is defined by Bjarnason et al. (2014) as a "*timeline of project events*", and it aids in starting discussions and ensures active participation. Factual evidence should be prepared beforehand and brought to the *retrospectives* (Lehtinen et al. 2017, Bjarnason et al.

2014). By distributing the factual evidence in advance, the participants are motivated, resulting in better discussions during the *retrospectives* (Bjarnason et al. 2014). In the case of evidence-based timelines, Bjarnason et al. (2014) argued that the facilitator should be responsible for the preparation and the summary of the outcome, thus avoiding any extra work for the remaining participants. To use factual evidence as input to the *retrospectives*, the relevant data must be easily accessible (Bjarnason et al. 2014).

Collaborating with relevant people outside team (C30)

Collaborating with relevant people outside the team is a way to deal with problems that are outside the team's control (C30), as addressed by Lehtinen et al. (2017) and Bjarnason et al. (2014). Through collaboration, important problems not solvable by the team alone can still be solved (Lehtinen et al. 2017). Inviting relevant external people to the *retrospectives* is a way to handle these problems right away (Bjarnason et al. 2014).

7.3 Results

In this subchapter, the results from the case study related to *retrospectives* will be presented and discussed. We will identify challenges and associated strategies, and the IDs in parentheses behind the strategies show what challenges they deal with. The results from each project will be addressed separately.

Project 1

I1 described how having *retrospectives* after each sprint was too frequent, considering the large time span of the project and its many sprints:

> But gradually, we have decided not to execute them regularly every sprint, but execute them more dynamically. Because when you have executed 30-40-50 sprints, you might start to run out of ... The need to execute retrospectives regularly, it slightly disappears. Instead, you address things when you see that "Now we need to have a talk about this". So, gradually, we ... we did not execute retrospectives after each sprint. (I1, P1)

On the other hand, another informant was a part of a sub-team in Project 1 that conducted *retrospectives* a few times every year. He found this to be too seldom. Hence, although a strategy can be to conduct *retrospectives* when necessary instead of after each sprint, a consequence can be that they are conducted too seldom. This strategy should therefore be used with care.

We discovered that *retrospectives* could easily end up becoming routine sessions with little or no valuable outcome. Two informants reported this challenge. Therefore, an informant from Project 1 suggested to be creative and vary the format, to make the *retrospectives* more interesting. One of the project teams based their discussions on drawings made by the team manager as a way to engage the participants. Another suggested and important strategy was to create a culture where participants dare to share their opinions.

Furthermore, two informants described difficulties in getting team members started and getting them to contribute. It was considered important to get team members to contribute in order to get good input. A suggested strategy for this challenge was to "*force people to contribute*". Contrarily, I9 shared how some found it challenging to always say something, indicating that forced contribution is not necessarily a good strategy:

> In the beginning, people found it difficult to say something. “Do I have to say something?” And I experienced that people were a bit tired of that kind of forced write-things-on-post-its-retrospectives. There were many who were tired of having to find three things that were good and three bad things every third week. That way, it was always positively received that I used a more loose format. (I9, P1)

Thus, changing the format can also enable more contribution, and perhaps be a better strategy than forcing everyone to contribute.

Conclusively, the following challenges were addressed in interviews with Project 1:

- C37: Executing too frequently
- C38: Executing too seldom
- C39: Lacking value creation for team members
- C8: Having passive team members

The following strategies were suggested in interviews with Project 1:

- Adjusting frequency (C37)
- Varying format (C39, C8)
- Building appropriate culture (C39)
- Making everyone contribute (C8)

Project 2

Project 2 sometimes experienced that problems recurred in *retrospectives*. These problems were affected by external factors and therefore outside the control of the team. The informants were not able to identify any strategies for these challenges.

I12 described how the *retrospectives* were perceived to be more valuable for managers than team members:

> There are many different types of retrospectives. I have tried to suggest that we should start something, but the boys like the “good and bad”. And also, I think that this is maybe something that one is more passionate about on a slightly higher level than ... I think the developers are like “Ah, time for retrospective again. Ah, okay, I guess we will have to participate”. (I12, P2)

I12 suggested trying different variants of *retrospectives* - e.g., use various analogies - to approach the problems in different ways. However, team members were not necessarily willing to try various formats, and varying the format was challenging to achieve when pushed on time.

Two informants from Project 2 reported unbalanced contributions from participants as a challenge. Both insinuated that personality traits heavily affected each participant’s contribution. The suggested strategy was to make everyone contribute - e.g., by using questioning techniques. I12 insinuated that an unbalanced contribution had to be accepted, but that each participant at least had to say one thing.

Conclusively, the following challenges were addressed in interviews with Project 2:

- C30: Dealing with problems outside the team’s control

- C34: Having recurring discussions
- C39: Lacking value creation for team members
- C8: Having passive team members

The following strategies were suggested in interviews with Project 2:

- Varying format (C39)
- Making everyone contribute (C8)

Project 3

In Project 3, they experienced that *retrospectives* were executed too frequently. An informant insinuated that this challenge emerged due to short sprints (lasting 2 weeks each) and the fact that the team members were only 40% involved in the project. The informant did not suggest any strategies for this challenge.

The lack of value creation for team members was also a challenge for Project 3. Sometimes, the discussions dragged out, as explained by I4:

> I think that if we have X number of problems to discuss in a meeting, we should maybe be a little time-efficient and not drag out on things that we see that "We will not get anywhere with this". (I4, P3)

Hence, the facilitation of the *retrospectives* appeared important to prevent discussions dragging out.

We found that it was challenging to extract good learning points in Project 3. Although not tested in the project, an informant believed that using various methods could be a good strategy for this challenge.

Through interviews, we also found that the project was struggling to find the appropriate level of detail, which indicates a lack of understanding of the practice. No strategies were identified.

Conclusively, the following challenges were addressed in interviews with Project 3:

- C37: Executing too frequently
- C39: Lacking value creation for team members
- C32: Lacking appropriate corrective actions
- C2: Lacking understanding of practice

The following strategies were suggested in interviews with Project 3:

- Having a skilled facilitator (C39)
- Varying format (C32)

Project 4

In Project 4, the team members that worked part-time did not fully see the value of spending that much time on *retrospectives*, as described by I5:

> The only thing that has been challenging is that some think that we spend too much time on it [retrospectives]. But that applies to those with a lot [other responsibilities] outside the team and who might not be that involved. (I5, P4)

We found that *retrospectives* can be challenging to facilitate. Furthermore, I5 shared the unbalanced contribution from the various participants:

> And there is always someone who does not contribute. It is the same ones who do not want to say anything. And then there are some who are very active. (I5, P4)

I5 also explained that no strategies were implemented for this challenge, as it should be up to each one to decide to what degree they wanted to contribute.

Due to COVID-19, the project was distributed during the execution of this study. The informant described how the participants possibly were affected negatively by the fact that the others' suggestions were visible to them on the digital board while writing down their own suggestions.

Conclusively, the following challenges were addressed in interviews with Project 4:

- C39: Lacking value creation for team members
- C8: Having passive team members
- C40: Facilitating
- C5: Working in distributed teams

No strategies were suggested in interviews with Project 4.

Project 5

In Project 5, team members commented on the high frequency of the *retrospectives* as they were all used to executing reflection sessions similar to *retrospectives* once a year due to the company culture surrounding the project.

I7 described the team members' lack of ownership and follow-up of corrective actions:

> And another challenge might be that people have little ownership of the things they want to improve. That it is often me [team manager] who have to nag things through. (I7, P5)

This indicates that it is important that all team members feel a sense of ownership of the retrospective outcome - not just the team manager. I7 tried to create ownership by making the various team members responsible for problems and the associated corrective actions.

As the project resources were not used to executing *retrospectives.* they also lacked an understanding of the practice. This was a hinder for Project 5.

In trying to create an appropriate culture that facilitated *retrospectives*, being distributed due to COVID-19 acted as a barrier, as explained by I7:

> When we [were co-located], people sat together, and then I felt that it was easier to get sort of "Ah, we are one team". But now that we are in teams that have never met each other [...]. Then I have the feeling it [the feeling of being one team] is slightly more challenging to obtain. (I7, P5)

Conclusively, the following challenges were addressed in interviews with Project 5:

- C37: Executing too frequently
- C35: Lacking follow-up of corrective actions
- C2: Lacking understanding of practice

- C5: Working in distributed teams

The following strategies were suggested in interviews with Project 5:

- Making team members responsible (C35)
- Building appropriate culture

Project 6

I8 shared that the corrective actions from *retrospectives* were not always followed-up sufficiently:

> Sometimes, I feel that we put in things, but then we go to the next point next time, and there is not necessarily a lot of follow-up ... Yes, there is some follow-up, but maybe not sufficient follow-up of whether we really have done what we said we would do and the status of that. (I8, P6)

When asked about solutions to this challenge, I8 replied that there were none at the moment.

Conclusively, the following challenge was addressed in interviews with Project 5:

- C35: Lacking follow-up of corrective actions

No strategies were suggested in interviews with Project 6.

Project 7

In Project 7, some teams seldom conducted *retrospectives*. This did not apply to the Scrum teams - i.e., the developer teams - but rather the business and architect teams. The solution was to define the *retrospectives* as deliveries, as explained by I14:

> They [business and architect teams] could have conducted more retrospectives. But we solved that in a way as well. We use products from PRINCE2, so we defined retrospectives as a sub-product to be delivered. (I14, P7)

Both informants from Project 7 described the lack of follow-up of corrective actions. I14 shared that they introduced *retrospectives* on the project level to ensure better follow-up:

> There was a reason why we introduced retrospectives on the project level and not just on the team level. And that was because we saw that not everything that emerged in the teams was transferred up in the system if that was necessary. [...] Because there was a lot that was - at least in the beginning - stopped at a certain level.

This indicates that the introduction of project-level *retrospectives* can be used as a strategy for the lack of follow-up of retrospective outcomes when the problems are outside the team's control or affect other parts of the project.

Lastly, we want to highlight that the teams of Project 7 sometimes utilized project resources from outside the team as facilitators. This indicates a focus on skilled facilitators, although no specific challenge seemed to be targeted.

Conclusively, the following challenges were addressed in interviews with Project 7:

- C38: Executing too seldom
- C35: Lacking follow-up of corrective actions
- C30: Dealing with problems outside the team's control

The following strategies were suggested in interviews with Project 7:

- Defining retrospectives as deliveries (C38)
- Introducing project-level retrospectives (C35, C30)
- Having a skilled facilitator

Project 8

No challenges or strategies were identified in Project 8.

Project 9

I18 described that problems outside the control of the team were time-consuming:

> Sometimes, we spend a lot of time on things that are outside our control. Things that do not work, but we are not able to do anything about it. Right? And then, there is no point in creating an action point on something outside your control. (I18, P9)

As a strategy for this challenge, I18 advocated focusing on the parts of the problem that the team actually could do something about. One should not spend time on uncontrollable elements.

Project 9 experienced that *retrospectives* could become boring or limit new input if the same fixed agenda was used every time. Therefore, they utilized a variety of tools - e.g., a game called *"Evil Mastermind"*. However, they had found that some tools were time-consuming and difficult to understand in the 30-60 minutes the project team spent on each *retrospective.* Thus, tools should be chosen with care.

Lastly, I16 felt that it was important to ensure follow-up of corrective actions:

> And in general, that you try to follow up when you have measures. Or else, it [retrospective] loses its purpose a bit. It can quickly become demotivating. (I16, P9)

Conclusively, the following challenges were addressed in interviews with Project 9:

- C30: Dealing with problems outside the team's control
- C39: Lacking value creation for team members
- C35: Lacking follow-up of corrective actions

The following strategies were suggested in interviews with Project 9:

- Focusing on controllable elements of problems (C30)
- Varying format (C39)
- Utilizing tools (C39)
- Introducing games (C39)

Project 10

I22 found it challenging to facilitate *retrospectives*:

> And I know from previous jobs, from my experience as a Scrum Master, that retrospective is one of the most challenging to facilitate. [...] But it [retrospective] is also what might ... it gives so much as well. But it is challenging to facilitate a retrospective. (I22, P10)

A skilled facilitator is therefore necessary.

Project 10 comprised one team of about 20 team members, hence making it difficult to execute *retrospectives.* As a result, the *retrospectives* ended up being time-consuming, and it was difficult to agree on corrective actions. A suggested strategy for this challenge was to divide the team into smaller sub-teams to execute the *retrospectives.* Tools could also have been utilized to gather ideas and vote on corrective actions.

Sometimes, there was a lack of appropriate corrective actions in Project 10, resulting in insufficient follow-up, as described by I20:

> But then there are some action points that we address that several of us might avoid doing. Because maybe we do not completely agree on them [...]. You look a little bit at the action points, but then there might be great disagreement or great uncertainty; thus, it might be smart to wait a sprint before one evaluates if it is necessary. (I20, P10)

On the other hand, we found that people sometimes forgot to execute the corrective actions, which was discovered at the next *retrospective.*

Project 10 also had trouble getting feedback from the more quiet and shy personalities, resulting in a lack of contribution. We found that making everyone contribute was not necessarily a good strategy, as the more passive participants end up agreeing with previous statements instead of adding something new to the discussions.

Two informants from Project 10 described challenges related to working distributed due to COVID-19. I22 explained that it was especially challenging to execute *retrospectives* virtually:

> I think that there are many other meetings that work fine on video. But retrospective is a very physical meeting, at least in my experience. (I22, P10)

Although no strategies were implemented in the project for this challenge, I22 proposed utilizing tools specifically designed for distributed teams.

Conclusively, the following challenges were addressed in interviews with Project 10:

- C40: Facilitating
- C41: Having too many participants
- C32: Lacking appropriate corrective actions
- C8: Having passive team members
- C35: Lacking follow-up of corrective actions
- C5: Working in distributed teams

The following strategies were suggested in interviews with Project 10:

- Having a skilled facilitator (C40)
- Dividing into smaller teams (C41)

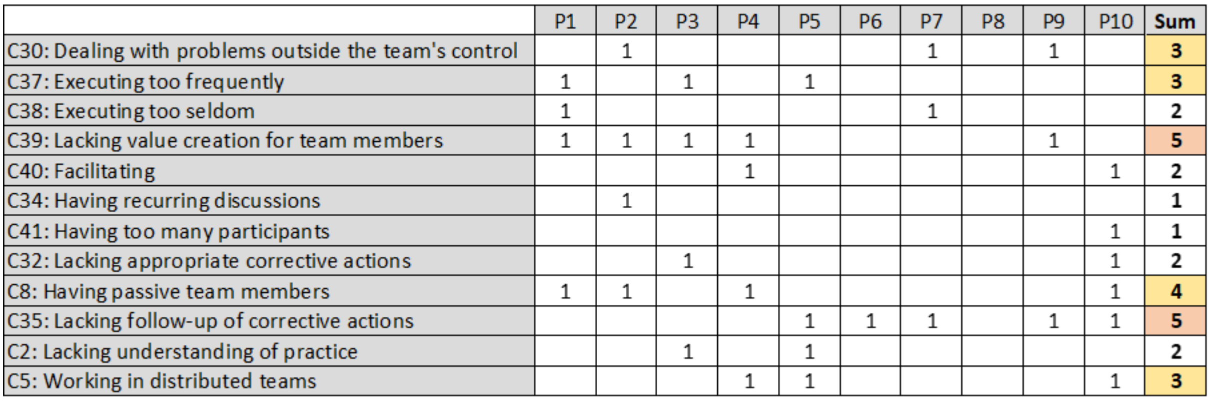
- Utilizing tools (C41, C5)

7.4 Discussion

In this subchapter, we will conduct a cross-case analysis where we discuss our findings and connect them to the identified related work. We will also answer our three research questions with respect to *retrospectives.*

Common challenges of *retrospectives* in Norwegian projects

We identified various challenges in interviews with the ten case projects. Figure 7.1 shows the number of projects where the challenges have been identified. Challenges found in three or more projects are considered common. The sum is yellow for challenges identified in 3-4 projects and red for those identified in five or more. The remaining challenges - with white sums - are considered uncommon. We will only discuss the common challenges.



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Sum
C30: Dealing with problems outside the team's control		1					1		1		**3**
C37: Executing too frequently	1		1		1						**3**
C38: Executing too seldom	1						1				**2**
C39: Lacking value creation for team members	1	1	1	1					1		**5**
C40: Facilitating				1						1	**2**
C34: Having recurring discussions		1									**1**
C41: Having too many participants										1	**1**
C32: Lacking appropriate corrective actions			1							1	**2**
C8: Having passive team members	1	1		1						1	**4**
C35: Lacking follow-up of corrective actions					1	1	1		1	1	**5**
C2: Lacking understanding of practice			1		1						**2**
C5: Working in distributed teams				1	1					1	**3**

Figure 7.1: Challenges of *Retrospectives* From the Case Study

The most common challenges were the lack of value creation for team members (C39) and the lack of follow-up of corrective actions (C35) - reported by five projects.

We found that fixed formats (Project 1, 2, and 9), unfocused discussions (Project 3), and part-time involvement in the project (Project 4) could reduce the team member's perceived value of the *retrospectives* (C39). The perceived value of *retrospectives* also seems higher for managers than for team members, as uncovered in Project 2.

The lack of follow-up of corrective actions (C35) can be caused by a lack of ownership of the problems and associated corrective actions, as found in Project 5. People can forget to follow-up (as in Project 6 and 10), or choose not to follow up (as in Project 7 and 10) - e.g., due to uncertainty and disagreements of the corrective actions. When people forget to follow up, this indicates a lack of appropriate follow-up routines. If the corrective actions are not followed up sufficiently, the *retrospectives* can become demotivating for the participants, as discovered in Project 9. This challenge is confirmed by Drury et al. (2012).

We identified the lack of contribution in five projects (C8). Contribution from all participants is considered important in order to get valuable input. The lack of contribution was often found to result from certain personality traits, as expressed by informants from Project 2, 4, and 10.

The challenge related to working in distributed teams (C5) emerged in this study due to COVID-19. As COVID-19 created a highly abnormal situation, we will not discuss this challenge further.

The last common challenges of *retrospectives* were the problems outside the control of the team (C30) and the high frequency (C37).

Problems affected by external factors lead to recurring discussions (Project 2) and are time-consuming (Project 9). In Project 7, they experienced that problems related to external factors were halted at a certain level and not brought to the project resources that could help solve the problems. The challenge was also discovered by Lehtinen et al. (2017).

Conducting *retrospectives* after each sprint can be perceived as too frequent (C37) if the project has many sprints (Project 1), short sprints (Project 3), or if the participants are involved part-time in the project (Project 3) or are new to the practice (Project 5).

Consequently, the common challenges of *retrospectives* in Norwegian projects (after disregarding C5) are the following:

- C39: Lacking value creation for team members
- C35: Lacking follow-up of corrective actions
- C8: Having passive team members
- C30: Dealing with problems outside the team's control
- C37: Executing too frequently

Strategies for the common challenges

Varying the format of *retrospectives* can be a good strategy for the lack of value creation for team members (C39), as suggested by informants from Project 1, 2, and 9. The facilitator should be creative to engage the participants and find ways to change their mindsets. This can be done through, e.g., basing discussions on drawings (Project 1), using analogies (Project 2), and tools such as games (Project 9). Jovanović et al. (2016) confirms the benefit of using games. Changing the format can also be a good strategy to increase contribution (C8). Nonetheless, one should be aware of new challenges related to varying the format; Participants can be reluctant to change (Project 2), limited time can make it difficult to realize the strategy (Project 2), and tools that are time-consuming and difficult to understand should be avoided (Project 9).

Building an appropriate culture can increase the value creation of the *retrospectives* (C39), as found in Project 1. The importance of creating an appropriate culture is also confirmed by Gupta et al. (2019). The participants must feel comfortable sharing in order to create value. Building culture can also enable *retrospectives* in general, without targeting a specific challenge, as in Project 5.

A skilled facilitator can increase the value creation in *retrospectives* (C39), e.g., by managing the time effectively (Project 3). The team can also benefit from having facilitators from outside the team, as in Project 7. The importance of a skilled facilitator is also addressed in the studies by Bjarnason et al. (2014) and Jovanović et al. (2016).

Making team members responsible for problems and associated corrective actions - as done in Project 5 - is a way to improve the follow-up of corrective actions (C35). Drury et al. (2012) also recommended making someone responsible for each corrective action, though not specifically team members.

The introduction of project-level *retrospectives* - i.e., *retrospectives* with all project resources present - can ensure a better follow-up of corrective actions (C35) and aid in dealing with problems outside the teams' control (C30). The strategy was used for the latter challenge in Project 7 and was perceived as effective. By arranging project-level *retrospectives*, project resources outside the team are involved directly in the discussions, enabling problem-solving. This ensures collaboration with relevant people outside the team, as addressed by Lehtinen et al. (2017) and Bjarnason et al. (2014). It also increases the range of controllable problems, making it easier to focus on controllable problems, which Lehtinen et al. (2017) reported as important. Although project-level *retrospectives* can prevent the outcome from being stopped at a certain level, the strategy in itself will not be sufficient to ensure a satisfactory level of follow-up of corrective actions (C35); Corrective actions from the project-level *retrospectives* can still remain unfulfilled - intentionally or unintentionally.

Making everyone contribute in the *retrospective* was a strategy utilized in both Project 1 and 2 to deal with passive team members (C8). However, more informants were sceptic or negative to this strategy than positive. Team members can find it challenging to have to share something (Project 1), some believe that contribution should be voluntary and not forced (Project 4), and the strategy can be ineffective as the inactive participants just agree on what has already been said (Project 10). Thus, we suggest to not explicitly make everyone contribute, but rather find a way to do it implicitly - e.g., by varying the format. Therefore, we recommend discarding the strategy of making everyone contribute.

A strategy to deal with problems outside the team's control (C30) is to focus on the controllable elements of the problems. This strategy is similar to focusing on controllable problems - as addressed by Lehtinen et al. (2017) - though slightly different. As discovered in Project 9, there can be controllable elements of a problem even though the problem is mainly outside the team's control. Focusing on the controllable elements prevents the team from spending time on uncontrollable elements, hence ensuring efficient *retrospectives*. However, the uncontrollable problem elements that project resources outside the team can solve should be brought to their attention and not be disregarded completely.

When dealing with too frequent execution of *retrospectives* (C37), adjusting the frequency - as done in Project 1 - is an obvious strategy. However, keeping the frequency dynamic by executing them when necessary can result in a new challenge; that they are executed too seldom (C38). We hence propose to establish a reduced but *fixed* frequency, to ensure that *retrospectives* are not forgotten or given a low priority.

We thus recommend the strategies presented in Table 7.3 for the common challenges of *retrospectives* in Norwegian projects.

Successful implementation of *retrospectives* in Norwegian projects

Based on the interviews and our own reflections, we recommend the following implementation of *retrospectives* in Norwegian projects:

- Vary the format - e.g., by utilizing various tools and games - to make the *retrospectives* more interesting and valuable. **Avoid** having a fixed agenda and format every time.
- Build a culture that makes everyone comfortable sharing their opinions.
- Invest time, training, and effort in the facilitation - e.g., by using facilitators from outside the team. **Avoid** taking the facilitation responsibility lightly, as *retrospectives* can be challenging to facilitate.
- Make team members responsible for problems and associated corrective actions to enable improved follow-up. **Avoid** making the team manager responsible for all follow-up.

Challenge	Strategy
C39: Lacking value creation for team members	Varying format Utilizing tools Introducing games Building appropriate culture Having a skilled facilitator
C35: Lacking follow-up of corrective actions	Making team members responsible Introducing project-level retrospectives
C8: Having passive team members	Varying format
C30: Dealing with problems outside the team's control	Introducing project-level retrospectives Focusing on controllable elements of problems Collaborating with relevant people outside team
C37: Executing too frequently	Establishing reduced and fixed frequency

Table 7.3: Strategies for *Retrospectives* From the Case Study

- Introduce *retrospectives* on several levels in the project - e.g., also on the project-level - to ensure effective problem-solving of problems not solvable by the team alone.
- Focus on the controllable elements of the problems. **Avoid** wasting time on uncontrollable elements in the *retrospectives.*
- Collaborate with relevant project resources outside the team to address the elements that are uncontrollable by the team.
- Establish an appropriate and fixed frequency to prevent too frequent or too seldom *retrospectives.* **Avoid** adjusting the frequency dynamically, as this can result in forgetting or giving a low priority to *retrospectives.*

8 Agile Practice: Stand-Up Meetings

This chapter will target the three research questions defined in Chapter 1 for the selected Agile practice *stand-up meetings*, following the research design described in Chapter 2. As we have chosen a nontraditional report structure that we consider to be beneficial for this study, we will present all related work, results, and discussion of *stand-up meetings* in this chapter. To make sure the reader understands the Agile practice under investigation in this chapter, we will commence with a short description of *stand-up meetings*.

8.1 Description of Practice

Stand-up meetings are short team meetings that are held frequently and where participants stand (Stray et al. 2016). The main objective of the *stand-up meetings* is to "*increase team awareness*" (Stray et al. 2016).

A standard agenda for *stand-up meetings* is:

1. What did you do yesterday?
2. What did you do today?
3. Are there any obstacles in your way?

In our study, we have chosen to make no distinction between *stand-up meetings* and "daily scrum". Daily scrums are meetings from the Scrum framework with almost the exact same characteristics as the *stand-up meetings*. As the two types of meetings are often used interchangeably, we believe it is reasonable to treat them as the same.

8.2 Related Work

In this subchapter, we will present literature related to *stand-up meetings* and the research questions. We have identified the related work by following the methods described in Subchapter 2.2.

8.2.1 Challenges

Challenges of *stand-up meetings* have been addressed by Stray et al. (2016), Stray et al. (2018), Moe et al. (2012), and Masood et al. (2020). These are listed in Table 8.1. The most frequently cited challenges are considered more applicable and robust and will therefore be discussed. The remaining challenges are considered self-explanatory and will not be discussed in further detail.

C39: Lacking value creation for team members

A challenge of *stand-up meetings* is that they can create little value for the team members (Stray et al. 2016, 2018, Moe et al. 2012). This challenge emerges when there is a lack of self-organization (Stray et al. 2016, 2018, Moe et al. 2012) or when there is a great mix of roles and seniority in the team (Stray et al. 2018). If the team lacks cross-functionality, it is challenging for team members to feel a connection to each other's work and pay attention (Stray et al. 2016, 2018, Moe et al. 2012). Similarly, managers can end up utilizing the *stand-up meetings* as status meetings to retrieve status information only relevant to themselves, providing little value to the team members (Stray et al. 2016, 2018). The status is retrieved by answering the first Scrum question. In both cases, the shared information and discussions are perceived as

ID	Challenge	References
C5	Working in distributed teams	Masood et al. (2020)
C37	Executing too frequently	Stray et al. (2016) Stray et al. (2018)
C39	Lacking value creation for team members	Stray et al. (2016) Stray et al. (2018) Moe et al. (2012)
C8	Having passive team members	Stray et al. (2016) Stray et al. (2018)
C41	Having too many participants	Stray et al. (2018)
C42	Taking too long	Stray et al. (2016) Stray et al. (2018)
C43	Lacking understanding of what to share	Stray et al. (2016) Stray et al. (2018) Moe et al. (2012)
C44	Disrupting other work	Stray et al. (2016) Stray et al. (2018)
C45	Reducing work hour flexibility	Stray et al. (2018)

Table 8.1: Challenges of *Stand-Up Meetings* From Literature

irrelevant for the team members, and the lack of relevancy poses challenges (Stray et al. 2018). When the *stand-up meetings* do not create value for the team members, they are regarded as "*boring*", as observed by Stray et al. (2016). Stray et al. (2018) argued that the lack of value creation for the team members was one of the primary challenges of *stand-up meetings*.

C43: Lacking understanding of what to share

It can be challenging to find the appropriate level of detail when sharing information in the *stand-up meetings* (Stray et al. 2016, 2018). While some team members go into too much detail only relevant for some of the participants (Moe et al. 2012, Stray et al. 2018), others barely share anything (Stray et al. 2016). This challenge is related to the status reporting when answering the first Scrum question about executed work (Stray et al. 2018).

8.2.2 Strategies

Strategies for *stand-up meetings* have been addressed by Stray et al. (2016), Stray et al. (2018), Pikkarainen et al. (2008), Moe et al. (2012), Gupta et al. (2019), and Babb et al. (2014). These are listed in Table 8.2. The IDs in parentheses show what challenges each strategy deals with, and those without parenthesis are enablers of the practice in general and do not target any specific challenges. The most frequently cited strategies are considered more applicable and robust and will therefore be discussed. However, as the studies by Stray et al. (2016) and Stray et al. (2018) are fairly similar, we do not consider the strategies reported by them to be robust unless also reported by other studies. The remaining strategies are considered self-explanatory and will not be discussed in further detail.

Having a clear and valuable objective (C39, C42, C43)

Stand-up meetings should have a clear objective that is valuable for all participants (Stray et al. 2016). Rather than being a status meeting valuable only for the manager (C39), the shared information should be new (Stray et al. 2018) and relevant for everyone (Stray et al. 2016). Stray et al. (2018) found that the participants valued the most the discussions not related to

Strategy	References
Implementing *self-organized team* (C39)	Stray et al. (2016)
Visualizing through tools (C39)	Stray et al. (2016) Stray et al. (2018)
Dividing in smaller teams (C41, C39, C8)	Stray et al. (2016) Stray et al. (2018)
Reminding through tools (C42)	Stray et al. (2016)
Using video calls when virtual (C5, C8)	Stray et al. (2016) Stray et al. (2018)
Standing up (C42)	Stray et al. (2016) Stray et al. (2018)
Having a clear and valuable objective (C39, C42, C43)	Stray et al. (2016) Stray et al. (2018) Pikkarainen et al. (2008)
Reporting status in other ways (C39)	Stray et al. (2018)
Focusing on tasks instead of individuals (C42, C41)	Stray et al. (2016)
Continuing discussions afterwards (C42, C43))	Stray et al. (2016) Moe et al. (2012)
Establishing reduced and fixed frequency (C37)	Stray et al. (2016) Stray et al. (2018)
Having a skilled facilitator (C42, C39, C8)	Stray et al. (2016) Stray et al. (2018)
Rotating facilitator (C39)	Stray et al. (2016) Stray et al. (2018)
Always starting and finishing on time (C42)	Stray et al. (2016)
Choosing most suitable meeting time (C44, C45, C42)	Stray et al. (2016) Stray et al. (2018)
Building appropriate culture	Gupta et al. (2019) Moe et al. (2012)
Assessing and adjusting format (C39)	Stray et al. (2016) Stray et al. (2018)
Microblogging on wiki page	Babb et al. (2014)

Table 8.2: Strategies for *Stand-Up Meetings* From Literature

the three standard Scrum questions, indicating a benefit from not strictly adhering to the Scrum questions.

One approach can be to focus solely on the future, thus eliminating the first question as defined by the Scrum framework - *"What have you done since we last met?"* - as it only provides value to the team manager and also is time-consuming (Stray et al. 2016, 2018). This would tackle the challenges of lacking value creation for team members (C39), taking too long (C42), and understanding what to share (C43). According to Stray et al. (2018), reducing the focus on executed work results in participants talking more to each other. Stray et al. (2016) proposed an agenda consisting of the following two questions: *"What will I do until we meet again to help our team achieve its goals?"* and *"What problems do I know of that may prevent progress?"*.

Another approach is to focus on problem-solving, as this is considered to bring value to the participants (Stray et al. 2016, 2018, Pikkarainen et al. 2008). This is achieved through the second and third Scrum question (Stray et al. 2018), which also advocates the elimination of the first Scrum question about executed work. Nonetheless, both Stray et al. (2016) and Stray et al. (2018) highlighted the challenge of actually reaching good decisions and solving the problems at the *stand-up meetings*, considering their short duration. Therefore, Stray et al. (2018) reasoned that easier problems could be solved in the *stand-up meetings*, but that more complex problems require a different forum.

Continuing discussions afterwards (C42, C43)

Arranging post meetings with relevant participants ensures the continuation of important discussions that are too long for the *stand-up meetings* (Stray et al. 2016). Hence, this is a useful strategy to not exceed the 15-minute time box (C42). Moe et al. (2012) also argued it was a useful strategy to deal with too detailed discussions not relevant for everyone (C43).

Building appropriate culture

Building an appropriate culture is also a strategy for *stand-up meetings* (Gupta et al. 2019, Moe et al. 2012). As with *retrospectives*, Gupta et al. (2019) reasoned that *stand-up meetings* were enabled by cultures promoting people, open interaction and communication, influence from team members, and profit-focus. On the other hand, Moe et al. (2012) advocated the importance of a culture focused on redundancy and solutions to process-related problems.

8.3 Results

In this subchapter, the results from the case study related to *stand-up meetings* will be presented and discussed. We will identify challenges and associated strategies, and the IDs in parentheses behind the strategies show what challenges they deal with. The results from each project will be addressed separately.

Project 1

A challenge with *stand-up meetings* is that they can become routine session with little or no value, as explained by I1:

> If you are not aware, it [stand-up meeting] too can become a routine session that does not give that much if you avoid talking about the hinders you have and that kinds of things. I know that some teams have chosen to conduct stand-ups 3 times a week instead of every day, for example. [...] Normally, not enough happened in a day to make it necessary to have stand-up every day. (I1, P1)

Hence, there was a lack of value creation due to too high frequency. Therefore, reducing the frequency is a strategy for both the high frequency and the lack of value creation. Another informant shared that it could become uninteresting for the team members if some or all were working on very different things, as they could only provide limited help to each other. Therefore, it seems that specialized team members - i.e., lack of *self-organized team* - reduce the value of *stand-up meetings.*

The *stand-up meetings* sometimes ended up taking too long due to talkative participants. A suggested strategy was to make everyone report written answers to the agenda questions in advance and use the *stand-up meetings* only to discuss barriers. Having a facilitator skilled at interrupting the participants was also perceived as important not to exceed the 15-minute time-box.

Because of COVID-19, Project 1 conducted virtual *stand-up meetings* as a distributed team. This made it possible to forget to attend the meetings.

Conclusively, the following challenges were addressed in interviews with Project 1:

- C39: Lacking value creation for team members
- C37: Executing too frequently
- C42: Taking too long
- C5: Working in distributed teams

The following strategies were suggested in interviews with Project 1:

- Establishing reduced and fixed frequency (C39, C37)
- Discussing barriers only (C42)
- Having a skilled facilitator (C42)

Project 2

Two informants from Project 2 reported that *stand-up meetings* could take too long. Sometimes, this was caused by detailed discussions not relevant for all participants. Other times, the status reporting could drag out due to talkative participants, as shared by I2:

> There are some who really like to talk, and then there can be more status and asking each other questions. (I2, P2)

In both cases, discussions were stopped and then continued after the *stand-up meetings*.

While working from home due to COVID-19, it was challenging for team members with small children also at home to attend the *stand-up meetings* as normal. To address this, the team implemented written execution of the *stand-up meetings*.

I12 expressed that it can be difficult to understand what level of detail should be shared in *stand-up meetings*:

> It is always a balance in stand-ups. To find out what is actually useful information. (I12, P2)

Conclusively, the following challenges were addressed in interviews with Project 2:

- C42: Taking too long
- C5: Working in distributed teams
- C43: Lacking understanding of what to share

The following strategies were suggested in interviews with Project 2:

- Continuing discussions afterwards (C42)
- Executing written *stand-up meetings* (C5)

Project 3

Two informants from Project 3 had experienced that too frequent *stand-up meetings* could be an issue. Team members can end up spending too much time in meetings since they are only working part-time in the project, as explained by I3:

> There are quite a lot of stand-ups or morning meetings around in the department, and we [team members] belong to various divisions and departments. [...] There is

> also something about trying to sort everything out, and not just have meeting after meeting. (I3, P3)

To prevent this challenge, Project 3 conducted 30-minute *stand-up meetings* once a week.

Sometimes, the team exceeded the time limit due to talkative team members. This was especially challenging when working distributed due to COVID-19. A strategy was to continue discussions afterwards. An informant also expressed the importance of having a skilled facilitator, as it can be difficult to interrupt and stop participants.

Lastly, the project lacked an understanding of what to share. It was challenging for them to know what to discuss in *stand-up meetings* versus other meetings. As several were familiar with long thematic meetings, they also struggled to target the discussions enough. The facilitator role was vital to deal with this.

Conclusively, the following challenges were addressed in interviews with Project 3:

- C37: Executing too frequently
- C42: Taking too long
- C5: Working in distributed teams
- C43: Lacking understanding of what to share

The following strategies were suggested in interviews with Project 3:

- Establishing reduced and fixed frequency (C37)
- Increasing duration (C37)
- Continuing discussions afterwards (C42)
- Having a skilled facilitator (C42, C43)

Project 4

I5 described the lack of attendance in the *stand-up meetings*:

> It's the attendance. Especially those that have a lot of other things.[...] So they [team members not attending] think that meetings are frequent. But at the same time, when we tried every other day, and when they did not attend one day, there were maybe four days between seeing them and hearing what was happening. And that doesn't work. So that's a small challenge. [...] It's the people from the business side that have full calendars, all the way, and meetings that coincide. (I5, P4)

Hence, the frequency of *stand-up meetings* can be too high, especially for part-time resources, as they have many other meetings and full calendars. This can result in them not attending at times. Reducing the frequency did not work well as a strategy. However, having full-time resources was suggested as a solution to the problem.

Several team members did not go into enough details when sharing, indicating a lack of understanding of the appropriate level of detail and a lack of contribution. This made it difficult for the rest to detect if the information was relevant for them in some way or not. To counter the challenge, the team asked follow-up questions to the less talkative participants.

Conclusively, the following challenges were addressed in interviews with Project 4:

- C46: Lacking attendance
- C37: Executing too frequently
- C43: Lacking understanding of what to share
- C8: Having passive team members

The following strategies were suggested in interviews with Project 4:

- Having full-time resources (C46, C37)
- Asking follow-up questions (C43, C8)

Project 5

Team members inexperienced with the practice can feel that *stand-up meetings* are a waste of time, as explained by I7:

> People think that it [stand-up meeting] is unnecessary use of time. People are like "What? Why should we have stand-up? We had sprint planning yesterday". (I7, P5)

The *stand-up meetings* could take too long because there was a lot to discuss. Although it was perceived difficult to stop discussions, continuing discussions after the *stand-up meetings* was used as a strategy.

I7 also thought that virtual *stand-up meetings* due to COVID-19 was a barrier, and that it was better to conduct them face-to-face.

Conclusively, the following challenges were addressed in interviews with Project 5:

- C39: Lacking value creation for team members
- C42: Taking too long
- C5: Working in distributed teams

The following strategies were suggested in interviews with Project 5:

- Continuing discussions afterwards (C42)

Project 6

In Project 6, the *stand-up meetings* sometimes took too long. I13 explained how this challenge occurred when hardware team members were present:

> Because those from the hardware gang, they dragged out in long discussions and wanted to discuss more. They probably weren't that good at being structured. (I13, P6)

The solution was to continue discussions afterwards.

The project also struggled with team members lacking the understanding of what to share in *stand-up meetings.* Several team members were inexperienced with the practice, and no one had clarified the expectations related to the agenda and the *stand-up meetings* in general. Thus, there was also a lack of understanding of the practice. The result was participants getting off topic and thus exceeding the 15-minute time-box.

Conclusively, the following challenges were addressed in interviews with Project 6:

- C42: Taking too long
- C43: Lacking understanding of what to share
- C2: Lacking understanding of practice

The following strategies were suggested in interviews with Project 6:

- Continuing discussions afterwards (C42)

Project 7

Sometimes, managers did not attend the *stand-up meetings* due to full calendars. Absent participants were required to inform of their absence and give a short written report in advance.

Another challenge was that the *stand-up meetings* exceeded the 15-minute time-box. The strategy was to interrupt discussions. I10 described it as a difficult strategy that should be used with care:

> We try to make people continue outside the meeting. [...] It's not always that easy. [...] But sometimes, it's smart to get things out as well. So to do everything according to processes and procedures isn't always that smart. Maybe you don't get what you need. Because you should remember why you stand there, it's to uncover things. (I10, P7)

Hence, spending more time in *stand-up meetings* can sometimes be worth it to enable discussions that can uncover important matters.

Furthermore, discussions could get into too much detail. This was also solved by continuing discussions after the *stand-up meetings*.

Conclusively, the following challenges were addressed in interviews with Project 7:

- C46: Lacking attendance
- C42: Taking too long
- C43: Lacking understanding of what to share

The following strategies were suggested in interviews with Project 7:

- Requiring notice of absence and written reports in advance (C46)
- Continuing discussions afterwards (C42, C43)

Project 8

In Project 8, there was a lack of understanding of *stand-up meetings* due to inexperience, as described by I11:

> Several had never participated in stand-up even though they had worked for 20 years. (I11, P8)

Conclusively, the following challenges were addressed in interviews with Project 8:

- C2: Lacking understanding of practice

No strategies were suggested in interviews with Project 8.

Project 9

I16 shared that the *stand-up meetings* could take too long when having too detailed discussions:

> Sometimes, we might see the need to discuss things more in detail. Then it [stand-up meeting] can drag out a little. (I16, P9)

The time went by especially fast when discussing details of tasks not relevant for everyone. A strategy for the challenges was to continue the discussions either right after the *stand-up meetings* or in separate meetings.

Conclusively, the following challenges were addressed in interviews with Project 9:

- C42: Taking too long
- C43: Lacking understanding of what to share

The following strategies were suggested in interviews with Project 9:

- Continuing discussions afterwards (C42, C43)

Project 10

The *stand-up meetings* in Project 10 lasted 30 minutes, as shared by I21:

> I have been taught that a stand-up should last 5-10 minutes, 10 minutes maximum, but these stand-ups last 30 minutes. That might be related to the fact that we are now at home offices, so it's not an actual stand-up. We are all sitting in chairs. (I21, P10)

This indicates that working distributed and sitting instead of standing causes the *stand-up meetings* to last longer. I21 also explained that they sometimes started to discuss solutions, which also lead to an increased duration. The strategy to stay within the time box was to continue discussions afterwards.

Conclusively, the following challenges were addressed in interviews with Project 10:

- C42: Taking too long
- C5: Working in distributed teams
- C43: Lacking understanding of what to share

The following strategies were suggested in interviews with Project 10:

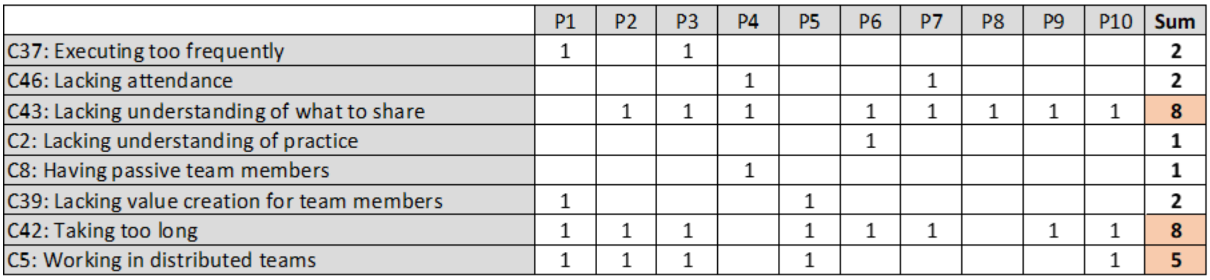
- Continuing discussions afterwards (C42)

8.4 Discussion

In this subchapter, we will conduct a cross-case analysis where we discuss our findings and connect them to the identified related work. We will also answer our three research questions with respect to *stand-up meetings.*

Common challenges of *stand-up meetings* in Norwegian projects

We identified various challenges in interviews with the ten case projects. Figure 8.1 shows the number of projects where the challenges have been identified. Challenges found in three or more projects are considered common. The sum is yellow for challenges identified in 3-4 projects and red for those identified in five or more. The remaining challenges - with white sums - are considered uncommon. We will only discuss the common challenges.



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Sum
C37: Executing too frequently	1		1								2
C46: Lacking attendance				1			1				2
C43: Lacking understanding of what to share		1	1	1		1	1	1	1	1	8
C2: Lacking understanding of practice						1					1
C8: Having passive team members				1							1
C39: Lacking value creation for team members	1				1						2
C42: Taking too long	1	1	1		1	1	1		1	1	8
C5: Working in distributed teams	1	1	1		1					1	5

Figure 8.1: Challenges of *Stand-Up Meetings* From the Case Study

The most common challenges were the lack of understanding of what to share (C43) and *stand-up meetings* taking too long (C42) - identified in eight of the ten projects.

Ten informants reported the lack of understanding of what to share (C43). Many projects struggled to find the right level of detail when reporting in the *stand-up meetings* (Project 2, 3, 4, 6, 7, 9, and 10). This challenge was confirmed by Stray et al. (2016), Stray et al. (2018) and Moe et al. (2012). It can be difficult to judge what information is useful to the remaining participants (Project 2) and what information should be shared in *stand-up meetings* versus other meetings (Project 3). Some share too many details (Project 3, 7, 9, and 10), whereas others share too little (Project 4). Considering the short duration of the *stand-up meetings*, it is not surprising that it is challenging to target the discussions and spend the time effectively.

Nine informants reported the challenge of *stand-up meetings* exceeding the 15-minute time box (C42), and it was also addressed by Stray et al. (2016) and Stray et al. (2018). This challenge can be caused by talkative participants (Project 1, 2 and 3), too detailed discussions (Project 2, 5, 6, 9 and 10), inexperience (Project 6), working distributed (Project 10), sitting instead of standing (Project 10), and lack of understanding of the practice (Project 6). The majority of the projects experienced that detailed discussions caused an increased duration, indicating that the challenge of too long *stand-up meetings* (C42) is closely related to the lack of understanding of what to share (C43). Naturally, these two challenges are reported by almost the same number of informants. It is also reasonable that it can be challenging to stay within the 15-minute time box when all participants are expected to report on several questions.

The challenge related to working in distributed teams (C5) emerged in this study due to COVID-19. As COVID-19 created a highly abnormal situation, we will not discuss this challenge further.

Consequently, the common challenges of *stand-up meetings* Norwegian projects (after disregarding C5) are the following:

- C43: Lacking understanding of what to share
- C42: Taking too long

Strategies for the common challenges

Stopping and then continuing certain discussions after the *stand-up meetings* is a common strategy to target the discussions (C43) and stay within the 15-minute time-box (C42). Stray et al. (2016) and Moe et al. (2012) also identified this strategy in their studies. The strategy was implemented in Project 2, 3, 5, 6, 7, 9, and 10. However, it can be challenging to stop discussions in practice (Project 3 and 5), making it necessary to have a skilled facilitator.

A skilled facilitator - also reported by Stray et al. (2016) and Stray et al. (2018) - can ensure that *stand-up meetings* do not take too long (C42) by targeting the discussions (C43). The facilitator is described as responsible for stopping too detailed or irrelevant discussion by informants from Project 1 and 3. At the same time, the facilitator should also let valuable discussions unfold (Project 7). Hence, the facilitator must understand what discussions are valuable for the participants.

For team members that share too few details in *stand-up meetings* (C43), it can be beneficial to ask follow-up questions, as done in Project 4. Nonetheless, as nine out of the ten informants reporting this challenge describe participants getting into too much detail rather than too little, we believe it would be misleading to recommend asking follow-up questions. We thus discard this as a good strategy for the lack of understanding of what to share.

Requiring written answers to the agenda questions in advance and only discussing barriers at the *stand-up meetings* is a strategy to stay within the 15-minute time-box (C42), as done in Project 1. This aligns with the findings by Stray et al. (2016) and Stray et al. (2018) to shift the focus from status reporting to future work and problem-solving. Quite interestingly, the guide for the Scrum framework has recently removed the suggested agenda questions for the meeting (Scrumguides.org 2020), indicating one should not necessarily strictly adhere to their original agenda.

We thus recommend the strategies presented in Table 8.3 for the common challenges of *stand-up meetings* in Norwegian projects.

Challenge	Strategy
C43: Lacking understanding of what to share	Continuing discussions afterwards Having a skilled facilitator
C42: Taking too long	Continuing discussions afterwards Having a skilled facilitator Having a clear and valuable objective Discussing barriers only

Table 8.3: Strategies for *Stand-Up Meetings* From the Case Study

Successful implementation of *stand-up meetings* in Norwegian projects

Based on the interviews and our own reflections, we recommend the following implementation of *stand-up meetings* in Norwegian projects:

- Continue too detailed discussions not relevant for everyone after the *stand-up meetings*, to stay within the 15-minute time-box and enable targeted discussions. Ensure that facilitators are skilled at stopping certain discussions by being knowledgeable of what discussions are considered valuable.

- Establish a clear and valuable objective - e.g., by acquiring written reports in advance and discussing barriers only at the *stand-up meetings* - to reduce the duration and increase the value creation. **Avoid** just going through the standard agenda for *stand-up meetings* without considering what creates the most value for all the participants.

9 Conclusion

9.1 Research Questions

This thesis supplements the existing body of knowledge by answering three research questions through a case study:

1. What are the common challenges of implementing the selected Agile practices in Norwegian projects?
2. What strategies can be implemented to deal with these common challenges?
3. How can the selected Agile practices be implemented successfully in Norwegian projects?

We selected the following five Agile practices for investigation: *self-organized team*, *customer involvement*, *release planning*, *retrospectives*, and *stand-up meetings.*

For the first research question, we identified common challenges. An overview of the common challenges is shown in Table 9.1. We include the comprehensive table for completeness, to clearly answer the research question, and to give the reader a good overview. The common challenges have also been discussed and listed practice-by-practice in their respective chapters. We recommend that practitioners in Norwegian projects make themselves familiar with the identified common challenges. This applies especially to the common challenges that span across several Agile practices and that have been identified in at least half of the case projects. The common challenges spanning across practices are: dealing with external dependencies, lacking understanding of practice, lacking sufficient *customer involvement*, and having passive team members. Additionally, the common challenges identified in at least half of the case projects are: lacking resources and competence, estimating tasks, obtaining a realistic backlog, lacking value creation for team members, lacking follow-up of corrective actions, lacking understanding of what to share, and taking too long.

For the second research question, we identified strategies for the common challenges. An overview of the strategies is shown in Table 9.2. We include the comprehensive table for completeness, to clearly answer the research question, and to give the reader a good overview. The strategies have also been listed practice-by-practice in their respective chapters. We recommend that practitioners in Norwegian projects make themselves familiar with the identified strategies. This applies especially to the strategies that deal with several challenges - also across several Agile practices. Identified strategies that deal with several challenges are: having full-time resources (team members and customer), appointing specific individuals as customer resources, training (team members and customer), increasing the authority of project teams, using customer proxy, having a skilled Product Owner, having a skilled facilitator, building appropriate culture, varying format, introducing project-level retrospectives, and continuing discussions afterward.

For the third research question, we formulated guidelines for each selected Agile practice. These guidelines were based on the answers to the first and second research questions. The guidelines for each practice can be found in their respective chapters. In Subchapter 9.2, we summarize our answers to the third research question.

Agile Practice	Common Challenges
Self-organized team	C1: Dealing with external dependencies C2: Lacking understanding of practice C4: Lacking resources and competence
Customer involvement	C14: Lacking sufficient customer involvement C13: Lacking clear requirements and needs C2: Lacking understanding of practice C15: Struggling to identify customer
Release planning	C1: Dealing with external dependencies C21: Estimating tasks C23: Obtaining realistic backlog C14: Lacking sufficient customer involvement C7: Lacking clear direction C8: Having passive team members
Retrospectives	C39: Lacking value creation for team members C35: Lacking follow-up of corrective actions C8: Having passive team members C30: Dealing with problems outside the team's control C37: Executing too frequently
Stand-up meetings	C43: Lacking understanding of what to share C42: Taking too long

Table 9.1: Common Challenges of Selected Agile Practices in Norwegian Projects

Agile Practice	Strategies
Self-organized team	Removing intermediaries (C1) Defining roles and responsibilities clearly (C2) Training (C2) Assigning mentors (C2) Having full-time resources (C4) Having equally available resources (C4) Acquiring a mix of competence (C4) Rotating roles (C4) Hiring external consultants (C4)
Customer involvement	Having full-time customer resources (C13, C14) Increasing the number of customer resources (C14) Appointing specific individuals as customer resources (C14, C15) Conducting more user tests (C14) Having customer resources as part of the project team (C14) Establishing close communication between customer and team members (C14) Training (C2) Arranging performance reviews (C2) Increasing the authority of project teams (C2, C13) Using customer proxy (C13, C15) Establishing task forces (C15)
Release planning	Highlighting task dependencies (C1) Dedicating time to release (C1) Defining expected release frequency (C1) Involving customer before finishing release (C1) Establishing guidelines for estimation (C21) Omitting task estimation (C21) Including tolerances in estimates (C21) Over-estimating (C21) Moving unfinished tasks to next iteration (C23) Limiting the scope of user stories (C23) Focusing on strictly necessary features (C23) Having a skilled Product Owner (C7, C23) Having a skilled facilitator (C8) Giving team members specific responsibilities (C8) Building appropriate culture (C8)
Retrospectives	Varying format (C39, C8) Utilizing tools (C39) Introducing games (C39) Building appropriate culture (C39) Having a skilled facilitator (C39) Making team members responsible (C35) Introducing project-level retrospectives (C30, C35) Focusing on controllable elements of problems (C30) Collaborating with relevant people outside team (C30) Establishing reduced and fixed frequency (C37)
Stand-up meetings	Continuing discussions afterwards (C42, C43) Having a skilled facilitator (C42, C43) Having a clear and valuable objective (C42) Discussing barriers only (C42)

Table 9.2: Strategies for Selected Agile Practices in Norwegian Projects

9.2 Implications

Below, we present our recommendations for a successful implementation of the selected Agile practices in Norwegian projects organized by who they apply to; the project management or the team. These recommendations are based on the identified strategies for the identified common challenges.

9.2.1 Strategies for the project management

Recruiting adequately, is one way the project management can ensure a more successful implementation of the selected Agile practices. We have discovered that an Agile team's success is highly dependant on the contributions from some specific roles. For *customer involvement* and *release planning*, having a customer representative as a part of the project team - e.g., as a Product Owner - is advantageous. Preferably, the Product Owner should have extensive knowledge of the customer's organization and be able to be in charge of task prioritization. Furthermore, the Product Owner should be a skilled communicator and be able to give the team members a sense of direction. Having a skilled meeting facilitator is decisive in obtaining efficient meetings. This applies especially to *retrospectives* and *stand-up meetings*, but also to *release planning* meetings. We recommend that the people filling key roles in Agile teams - i.e., as meeting facilitators, Scrum Masters, Product Owners, and customer resources - are recruited with both their personal and technical skills in mind.

Teams should be put together to inherit a mix of competence - e.g., both seniors and juniors, and both developers and customer representatives. Including customer representatives in the project team seems to be particularly important, as this is requested in many of the projects we have studied. Preferably, customer representatives are acquired full-time for the project - at least in medium and large-scale projects. The project management and the customer should agree on a level of customer participation, adequate for both parts, before starting the project. Acquiring a sufficient number of customer resources in a project is beneficial for *customer involvement* and *release planning.*

Training the team in the Agile practice has come up as a strategy for tackling various challenges. Our findings show that Project Managers struggle to communicate clearly why the project has chosen an Agile approach and what is expected of the team members and team managers. Having defined roles and responsibilities has come up several times, including how to behave in a *self-organizing team* and the expected release frequency. Assigning mentors for new team members to increase their understanding of how to work in a *self-organizing team* is one way to do the training. Training team members in the customer's domain can also make the team more self-organized and thereby more efficient. We conclude that most Agile project teams can benefit from over-clarifying what is expected of the project team. If the project resources lack the necessary competence, the project management should consider hiring external consultants. However, we suspect that increasing training and explicitly conveying expectations are the easiest ways to maximize value with minimum effort.

Regarding full-time versus part-time project resources, our findings do not indicate that one option is significantly better than the other. Some projects have shown to cope well with part-time team members, whereas others have not. We have seen a tendency to cope better with part-time resources for organizations already used to working on several projects in parallel. However, our findings indicate that equal involvement of resources is beneficial - i.e., that the resources work on the project simultaneously. We identified this strategy for *self-organized team.* However, it is reasonable to believe that this also applies to *stand-up* *meetings*, *retrospectives*, and *release planning* meetings, as the strategy was addressed with regard to meetings. Additionally, having full-time project resources - either team members or customer resources in the project - has been reported to tackle several challenges (C4, C13, C14). Consequently, we suspect that it might be slightly less complicated to manage a project with only full-time employees.

Developing a culture benefiting the Agile team is also essential. We have found that the implementation of Agile practices benefits from having a culture of trust. Team members should feel comfortable voicing their opinions and contributing to meetings. This positively affects both *retrospectives* and *release planning.*

Training the customer resources in the Agile frameworks in projects will increase the value of their contribution. This applies to customers who are actively involved in the project team. We acknowledge that the strategy is not easily implemented with external customers - especially so if the customers are from a different organization than the project management. In our study, we have identified that the customer often lacks an understanding of their responsibilities. Training the customer in the Agile planning procedures and how the customer can best contribute is beneficial for *release planning.* This should include clarification of what is expected of the customer and training in how to specify requirements clearly. Additionally, teaching the customer about the value of their contribution can increase *customer involvement.* Lastly, the customer resources must be chosen with both their communicative and technical skills in mind.

Compensating for the lack of *customer involvement* is also possible. For example, one can introduce the project participant and relevant field experts. When project participants and customer resources know each other well, the need for organized communication decreases. Arranging team-building events and facilitating direct and frequent communication can be beneficial. Furthermore, increasing the team's authority can decrease the necessary availability of a customer resource. Lastly, if sufficient *customer involvement* from actual customer representatives is not possible, using a customer proxy could be an option.

9.2.2 Strategies for the team

The team can also contribute in various ways to improve the implementation of the selected Agile practices in Norwegian projects. In this subchapter, we present several strategies for the team.

The team can contribute to **avoiding delays**, caused by both external and internal factors. External interruptions outside of the team's control will occur throughout the project. In our study, we have identified several ways the team can compensate for these interruptions. First, adding slack in the estimates - e.g., through over-estimating and introducing tolerances - is an easy and effective way to prepare for the delays caused by external factors. Second, making a register of tasks with external dependencies and working on them as early as possible has shown to be effective. Third, the team can improve estimation accuracy by involving customers in estimations. This way, dependencies are uncovered, and the detail of tasks specification increases. Fourth, focusing on strictly necessary features will decrease the probability of unwanted scope creep. Lastly, lack of *customer involvement* can also cause delays. Adding the customer to a digital communication platform - e.g., Slack - can increase informal communication and decrease time spent waiting for clarifications.

Increasing the value of meetings is also something the team can impact. Firstly, contributing to a trusting and active culture must be the responsibility of all individuals. The *retrospectives* can be both more valuable and enjoyable if the format is varied. Using digital tools and introducing games to the agenda can be beneficial. Also, focusing on controllable elements of problems will increase the value of the meeting. Additionally, making team members responsible for follow-ups will increase their ownership and the possibility of improvements. We have observed that it often can be beneficial to reduce the frequency of *retrospectives* if they seem unnecessary every iteration. Nevertheless, the frequency should be fixed to avoid them from being completely forgotten or downgraded. Regarding *stand-up meetings*, keeping them short and concise can be challenging. Firstly, all teams should define a clear and valuable objective for their meeting. Secondly, the team should continue longer discussions after the *stand-up meetings* have ended. Lastly, the team should consider only discussing barriers, as the barriers are normally the most important of the *stand-up meetings*.

10 Limitations and Further Work

10.1 Limitations

This study has to be seen in the light of some limitations. First, we investigated a variety of project types. Our analysis is based on what challenges and strategies are common across all these project types. We did not conduct our analysis based on project attributes. Therefore, our findings must be considered a generalization across project types.

Second, the study might be prone to some sample bias. A majority of the case projects were sampled through a research project at NTNU and a consultant company specializing in Agile PM models. It is natural to suspect that these companies' interest in optimizing their Agile implementation is above average. Therefore, our sample might not be representative of all Norwegian projects.

Last, our data analysis process can not be replicated by outsiders. For ethical reasons, and due to confidentiality agreements, our interview transcripts can not be made public. Therefore, only our supervisors and ourselves can replicate the data analysis process in its entirety.

In addition to the above, deficiencies in the data collection and analysis are described in Subchapter 2.4.

10.2 Further Work

In this study, we have identified the most common challenges, but we have not focused especially on their impacts. Therefore, we propose to investigate which of the identified common challenges have the most influential negative impact and how they can be mitigated to a greater extent.

Some of the strategies can be challenging to implement. Training is a strategy that practitioners can use for various challenges, but we have found that it is difficult to implement successfully. As a strategy with a great potential for value-creation, we suggest to do further research on how to train project resources to enable the successful implementation of Agile practices. Omitting estimation is another strategy that can be challenging to implement. Although it is easy for the team to skip task estimation, the strategy poses challenges for project management and the stakeholders, who lose an important control element. We would like to see further research on how to best deal with the time-consuming estimations providing little value for the team members and on the feasibility of omitting estimation completely. We also suggest to further investigate the implementation challenges of the various strategies, focusing on the strategies perceived to be especially effective.

The benefits of some strategies are uncertain. We have not been able to conclude if the suggested strategy of having full-time resources is in fact better than having part-time resources. We propose to investigate whether full-time resources really make the implementation of Agile practices substantially better than part-time resources. Another uncertain strategy is the involvement of customer resources as a part of the project team. Although the involvement of customer resources as a part of the project team has been identified as a strategy, we have also seen that it is possible to achieve success with external customer resources. We propose to do more research on what involvement of the customer resources is the best and investigate further the success factors of projects with external customers.

We also suggest to replicate the study, but for other Agile practices and other projects.

References

Ayed, H., Vanderose, B. & Habra, N. (2017), Agile cultural challenges in europe and asia: insights from practitioners, *in* '2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)', IEEE, pp. 153–162.

Babb, J., Hoda, R. & Nørbjerg, J. (2014), 'Embedding reflection and learning into agile software development', *IEEE software* **31**(4), 51–57.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R. et al. (2001), 'Manifesto for agile software development', `https://agilemanifesto.org/`. Accessed 18.05.2021.

Bernsmed, K. & Jaatun, M. G. (2019), Threat modelling and agile software development: Identified practice in four norwegian organisations, *in* '2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)', IEEE, pp. 1–8.

Bjarnason, E., Hess, A., Svensson, R. B., Regnell, B. & Doerr, J. (2014), 'Reflecting on evidence-based timelines', *IEEE software* **31**(4), 37–43.

Booth, A., Sutton, A. & Papaioannou, D. (2016), *Systematic approaches to a successful literature review*, Sage.

Brooks, J., McCluskey, S., Turley, E. & King, N. (2015), 'The utility of template analysis in qualitative psychology research', *Qualitative research in psychology* **12**(2), 202–222.

Cao, L., Mohan, K., Xu, P. & Ramesh, B. (2009), 'A framework for adapting agile development methodologies', *European Journal of Information Systems* **18**(4), 332–343.

Cao, L. & Ramesh, B. (2008), 'Agile requirements engineering practices: An empirical study', *IEEE software* **25**(1), 60–67.

Cervone, H. F. (2011), 'Understanding agile project management methods using scrum', *OCLC Systems & Services: International digital library perspectives* .

Crowston, K., Li, Q., Wei, K., Eseryel, U. Y. & Howison, J. (2007), 'Self-organization of teams for free/libre open source software development', *Information and software technology* **49**(6), 564–575.

Dingsøyr, T., Nerur, S., Balijepally, V. & Moe, N. B. (2012), 'A decade of agile methodologies: Towards explaining agile software development'.

Drury, M., Conboy, K. & Power, K. (2012), 'Obstacles to decision making in agile software development teams', *Journal of Systems and Software* **85**(6), 1239–1254.

Dybå, T. & Dingsøyr, T. (2008), 'Empirical studies of agile software development: A systematic review', *Information and software technology* **50**(9-10), 833–859.

Dybå, T. & Dingsøyr, T. (2015), Agile project management: From self-managing teams to large-scale development, *in* '2015 IEEE/ACM 37th IEEE International Conference on Software Engineering', Vol. 2, IEEE, pp. 945–946.

Fernandez, D. J. & Fernandez, J. D. (2008), 'Agile project management—agilism versus traditional approaches', *Journal of Computer Information Systems* **49**(2), 10–17.

Golfarelli, M., Rizzi, S. & Turricchia, E. (2013), 'Multi-sprint planning and smooth replanning: An optimization model', *Journal of systems and software* **86**(9), 2357–2370.

Grapenthin, S., Poggel, S., Book, M. & Gruhn, V. (2015), 'Improving task breakdown comprehensiveness in agile projects with an interaction room', *Information and Software Technology* **67**, 254–264.

Gupta, M., George, J. F. & Xia, W. (2019), 'Relationships between it department culture and agile software development practices: An empirical investigation', *International Journal of Information Management* **44**, 13–24.

Heikkilä, V. T., Paasivaara, M., Rautiainen, K., Lassenius, C., Toivola, T. & Järvinen, J. (2015), 'Operational release planning in large-scale scrum with multiple stakeholders–a longitudinal case study at f-secure corporation', *Information and Software Technology* **57**, 116–140.

Highsmith, J. & Cockburn, A. (2001), 'Agile software development: The business of innovation', *Computer* **34**(9), 120–127.

Hoda, R. (2011), 'Self-organizing agile teams: A grounded theory'.

Hoda, R. & Murugesan, L. K. (2016), 'Multi-level agile project management challenges: A self-organizing team perspective', *Journal of Systems and Software* **117**, 245–257.

Hoda, R., Noble, J. & Marshall, S. (2008), Agile project management, *in* 'New Zealand Computer Science Research Student Conference, NZCSRC 2008'.

Hoda, R., Noble, J. & Marshall, S. (2011), 'The impact of inadequate customer collaboration on self-organizing agile teams', *Information and Software Technology* **53**(5), 521–534.

Jovanović, M., Mesquida, A.-L., Radaković, N. & Mas, A. (2016), 'Agile retrospective games for different team development phases', *Journal of Universal Computer Science* **22**(12), 1489–1508.

Kakar, A. K. (2017), 'Assessing self-organization in agile software development teams', *Journal of computer information systems* **57**(3), 208–217.

Kruchten, P. (2004), 'Scaling down large projects to meet the agile sweet spot', *IBM developerWorks* **13**.

Layman, L., Williams, L., Damian, D. & Bures, H. (2006), 'Essential communication practices for extreme programming in a global software development team', *Information and software technology* **48**(9), 781–794.

Lehtinen, T. O., Itkonen, J. & Lassenius, C. (2017), 'Recurring opinions or productive improvements—what agile teams actually discuss in retrospectives', *Empirical Software Engineering* **22**(5), 2409–2452.

Marshall, C. & Rossman, G. B. (2014), *Designing qualitative research*, Sage publications.

Marshall, C. & Rossman, G. B. (2016), *Designing qualitative research*, 6 edn, Sage publications.

Masood, Z., Hoda, R. & Blincoe, K. (2020), 'How agile teams make self-assignment work: a grounded theory study', *Empirical Software Engineering* **25**(6), 4962–5005.

Metier OEC (2021), 'Kort om prince2 agile®', `https://www.prince2.no/om-prince2-agile/`. Accessed 02.06.2021.

Moe, N. B., Aurum, A. & Dybå, T. (2012), 'Challenges of shared decision-making: A multiple case study of agile software development', *Information and Software Technology* **54**(8), 853–865.

Mohammadi, S., Nikkhahan, B. & Sohrabi, S. (2008), An analytical survey of" on-site customer" practice in extreme programming, *in* 'International Symposium on Computer Science and its Applications', IEEE, pp. 1–6.

NSD (2021), 'Data protection services', `https://www.nsd.no/en/data-protection-services`. Accessed 1.2.21.

NTNU (2021*a*), 'Datainnsamling', `https://innsida.ntnu.no/wiki/-/wiki/Norsk/datainnsamling`. Accessed 1.2.21.

NTNU (2021*b*), 'Finding sources', `https://innsida.ntnu.no/wiki/-/wiki/English/How+to+search+for+literature`. Accessed 2.5.21.

NTNU (2021*c*), 'Informasjonsklassifisering - informasjonssikkerhet', `https://innsida.ntnu.no/wiki/-/wiki/Norsk/Informasjonsklassifisering+-+informasjonssikkerhet`. Accessed 1.2.21.

NTNU (2021*d*), 'Lagring av filer og dokumenter', `https://innsida.ntnu.no/wiki/-/wiki/Norsk/Lagring+av+filer+og+dokumenter`. Accessed 1.2.21.

NTNU (2021*e*), 'Lagringsguide', `https://innsida.ntnu.no/wiki/-/wiki/Norsk/lagringsguide`. Accessed 1.2.21.

NTNU (2021*f*), 'Skjemaverktoy', `https://innsida.ntnu.no/wiki/-/wiki/Norsk/Skjemaverkt\OT1\oy`. Accessed 1.2.21.

NTNU University Library (2021), 'Nvivo introduction for phd candidates and researchers [webinar]'. 02.03.21-03.03.21.

Nuottila, J., Aaltonen, K. & Kujala, J. (2016), 'Challenges of adopting agile methods in a public organization', *International Journal of Information Systems and Project Management* **4**(3), 65–85.

Papatheocharous, E. & Andreou, A. S. (2014), 'Empirical evidence and state of practice of software agile teams', *Journal of Software: Evolution and Process* **26**(9), 855–866.

Patton, M. Q. (2002), *Qualitative Research Evaluation Methods*, 3 edn, Sage.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P. & Still, J. (2008), 'The impact of agile practices on communication in software development', *Empirical Software Engineering* **13**(3), 303–337.

PMI (2017), 'Pmbok® guide – sixth edition', `https://www.pmi.org/pmbok-guide-standards/foundational/pmbok`. Accessed 02.06.2021.

Ramesh, B., Cao, L. & Baskerville, R. (2010), 'Agile requirements engineering practices and challenges: an empirical study', *Information Systems Journal* **20**(5), 449–480.

Saeeda, H., Dong, J., Wang, Y. & Abid, M. A. (2020), 'A proposed framework for improved software requirements elicitation process in scrum: Implementation by a real-life norway-based it project', *Journal of Software: Evolution and Process* **32**(7), e2247.

Saldana, J. (2013), 'Coding manual for qualitative analysis'.

Sandstø, R. & Reme-Ness, C. (2021), 'Agile practices and impacts on project success.', *Journal of Engineering, Project, and Production Management* **11**(3), 255.

Scrumguides.org (2020), 'Changes between 2017 and 2020 scrum guides', `https://scrumguides.org/revisions.html`. Accessed 09.06.2021.

Scrum.org (2021), 'What is a sprint retrospective?', `https://www.scrum.org/resources/what-is-a-sprint-retrospective`. Accessed 09.06.2021.

Serrador, P. & Pinto, J. K. (2015), 'Does agile work?—a quantitative analysis of agile project success', *International Journal of Project Management* **33**(5), 1040–1051.

Siddique, L. & Hussein, B. A. (2014), Practical insight about risk management process in agile software projects in norway, *in* '2014 IEEE International Technology Management Conference', IEEE, pp. 1–4.

Siddique, L. & Hussein, B. A. (2017), 'Grounded theory study of the contracting process in agile projects in norway's software industry', *Examination of the challenges in agile projects from the suppliers 'perspective in Norways software industry Insight and recommendations* p. 65.

Sidky, A., Arthur, J. & Bohner, S. (2007), 'A disciplined approach to adopting agile practices: the agile adoption framework', *Innovations in systems and software engineering* **3**(3), 203–216.

Stray, V., Moe, N. B. & Sjoberg, D. I. (2018), 'Daily stand-up meetings: start breaking the rules', *IEEE Software* **37**(3), 70–77.

Stray, V., Sjøberg, D. I. & Dybå, T. (2016), 'The daily stand-up meeting: A grounded theory study', *Journal of Systems and Software* **114**, 101–124.

Tanner, M. & Mackinnon, A. (2015), 'Sources of interruptions experienced during a scrum sprint', *Electronic Journal of Information Systems Evaluation* **18**(1), 3.

UiO (2021), 'Nettskjema', `https://nettskjema.no/user/index.html`. Accessed 1.2.21.

Unger-Windeler, C., Klünder, J. & Schneider, K. (2019), A mapping study on product owners in industry: identifying future research directions, *in* '2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)', IEEE, pp. 135–144.

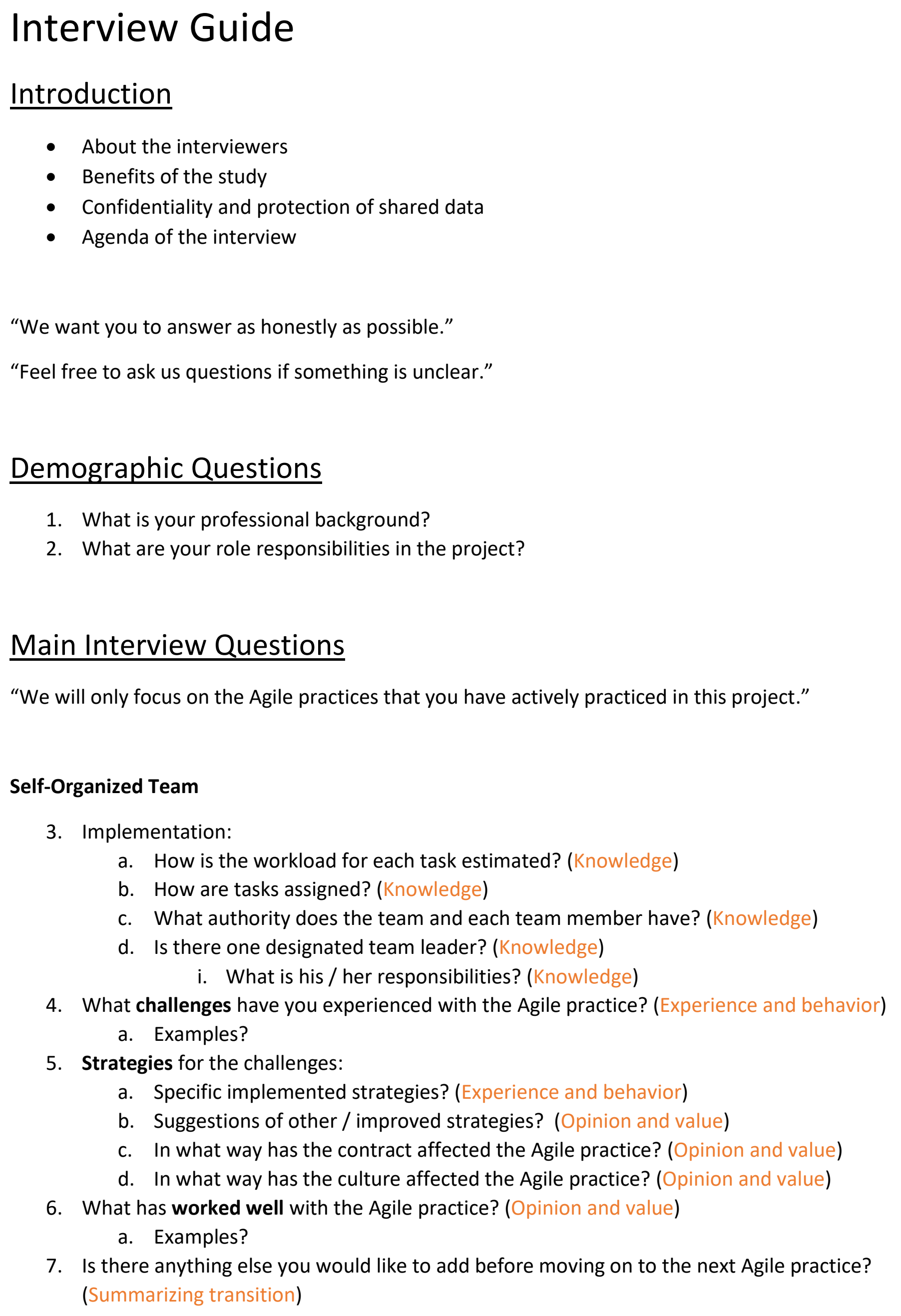
Vanhala, E. & Kasurinen, J. (2019), The role of the customer in an agile project: A multi-case study, *in* 'International Conference on Software Business', Springer, pp. 208–222.

VersionOne (2020), '14th annual state of agile report', `https://stateofagile.com/#`. Accessed 18.05.2021.

Werder, K. & Maedche, A. (2018), 'Explaining the emergence of team agility: a complex adaptive systems perspective', *Information Technology & People* .

Yin, R. K. (2003), *Case study research: Design and Methods*, Vol. 5, 3 edn, Sage publications.

A Interview Guide



Interview Guide

Introduction

- About the interviewers
- Benefits of the study
- Confidentiality and protection of shared data
- Agenda of the interview

"We want you to answer as honestly as possible."

"Feel free to ask us questions if something is unclear."

Demographic Questions

1. What is your professional background?
2. What are your role responsibilities in the project?

Main Interview Questions

"We will only focus on the Agile practices that you have actively practiced in this project."

Self-Organized Team

3. Implementation:
    a. How is the workload for each task estimated? (Knowledge)
    b. How are tasks assigned? (Knowledge)
    c. What authority does the team and each team member have? (Knowledge)
    d. Is there one designated team leader? (Knowledge)
        i. What is his / her responsibilities? (Knowledge)
4. What **challenges** have you experienced with the Agile practice? (Experience and behavior)
    a. Examples?
5. **Strategies** for the challenges:
    a. Specific implemented strategies? (Experience and behavior)
    b. Suggestions of other / improved strategies? (Opinion and value)
    c. In what way has the contract affected the Agile practice? (Opinion and value)
    d. In what way has the culture affected the Agile practice? (Opinion and value)
6. What has **worked well** with the Agile practice? (Opinion and value)
    a. Examples?
7. Is there anything else you would like to add before moving on to the next Agile practice? (Summarizing transition)

Release Planning

3. Implementation:
    a. Who attends? (Knowledge)
    b. What is the agenda? (Knowledge)
    c. What tools / techniques are used? (Knowledge)
    d. How are requirements / tasks prioritized? (Knowledge)
4. What **challenges** have you experienced with the Agile practice? (Experience and behavior)
    a. Examples?
5. **Strategies** for the challenges:
    a. Specific implemented strategies? (Experience and behavior)
    b. Suggestions of other / improved strategies? (Opinion and value)
    c. In what way has the contract affected the Agile practice? (Opinion and value)
    d. In what way has the culture affected the Agile practice? (Opinion and value)
6. What has **worked well** with the Agile practice? (Opinion and value)
    a. Examples?
7. Is there anything else you would like to add before moving on to the next Agile practice? (Summarizing transition)

Customer Involvement

3. Implementation:
    a. Who is your customer? (Knowledge)
    b. What tasks do the customer have in the project? (Knowledge)
    c. Do developers understand the customer's requirements? (Opinion and value)
    d. Do you believe that the customer is involved "enough"? (Opinion and value)
4. What **challenges** have you experienced with the Agile practice? (Experience and behavior)
    a. Examples?
5. **Strategies** for the challenges:
    a. Specific implemented strategies? (Experience and behavior)
    b. Suggestions of other / improved strategies? (Opinion and value)
    c. In what way has the contract affected the Agile practice? (Opinion and value)
    d. In what way has the culture affected the Agile practice? (Opinion and value)
6. What has **worked well** with the Agile practice? (Opinion and value)
    a. Examples?
7. Is there anything else you would like to add before moving on to the next Agile practice? (Summarizing transition)

Retrospectives

3. Implementation:
    a. Who attends? (Knowledge)
    b. What is the agenda? (Knowledge)
    c. What tools / techniques are used? (Knowledge)
    d. What is the outcome? (Knowledge)
    e. How is this outcome utilized afterwards? (Experience and behavior)
4. What **challenges** have you experienced with the Agile practice? (Experience and behavior)

   a. Examples?
5. **Strategies** for the challenges:
   a. Specific implemented strategies? (Experience and behavior)
   b. Suggestions of other / improved strategies? (Opinion and value)
   c. In what way has the contract affected the Agile practice? (Opinion and value)
   d. In what way has the culture affected the Agile practice? (Opinion and value)
6. What has **worked well** with the Agile practice? (Opinion and value)
   a. Examples?
7. Is there anything else you would like to add before moving on to the next Agile practice? (Summarizing transition)

Stand-Up Meetings

3. Implementation:
   a. Who attends? (Knowledge)
   b. What is the agenda? (Knowledge)
4. What **challenges** have you experienced with the Agile practice? (Experience and behavior)
   a. Examples?
5. **Strategies** for the challenges:
   a. Specific implemented strategies? (Experience and behavior)
   b. Suggestions of other / improved strategies? (Opinion and value)
   c. In what way has the contract affected the Agile practice? (Opinion and value)
   d. In what way has the culture affected the Agile practice? (Opinion and value)
6. What has **worked well** with the Agile practice? (Opinion and value)
   a. Examples?
7. Is there anything else you would like to add before moving on to the next Agile practice? (Summarizing transition)

Ending

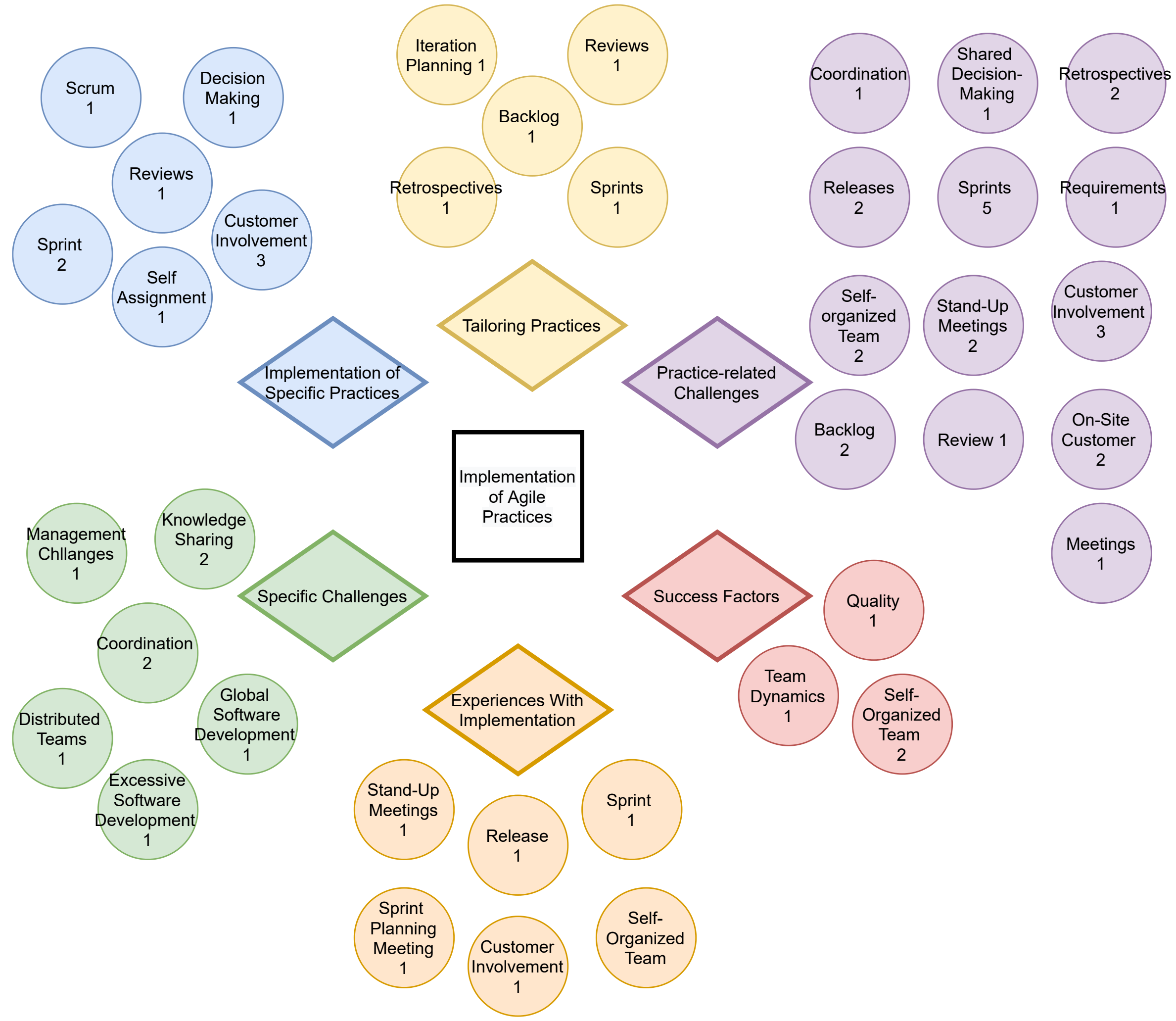
8. Do you have anything to add?
9. Do you have any questions for us?

What is next?

- Transcription clarifications
- Report finished in June.

"Thank you for your time!"

B Mind Map for Exclusion of Literature



The numbers inside the circles represent the number of identified references.

Cornelia Reme-Ness and Rebecca Sandstø

Successful Implementation of Agile Practices in Norwegian Projects: A Case Study

NTNU
Norwegian University of
Science and Technology