Master's thesis

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

Juni Leirvik Larsen

Remote Onboarding of Newly Graduated Software Developers to Agile Teams: A Multi-Case Study

Exploring Onboarding Strategies, Use of Agile Practices & Activities in Onboarding, and Success Factors of Onboarding in Large Norwegian IT-Consultant Companies

Master's thesis in Computer Science
Supervisor: Torgeir Dingsøyr
June 2023

NTNU
Norwegian University of
Science and Technology

Juni Leirvik Larsen

Remote Onboarding of Newly Graduated Software Developers to Agile Teams: A Multi-Case Study

Exploring Onboarding Strategies, Use of Agile Practices & Activities in Onboarding, and Success Factors of Onboarding in Large Norwegian IT-Consultant Companies

Master's thesis in Computer Science
Supervisor: Torgeir Dingsøyr
June 2023

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

NTNU
Norwegian University of
Science and Technology

Abstract

The motivation behind this study originates from the recent increase in remote software development teams, the need to provide a warm welcome for newly graduated developers and the importance of high-quality communication within agile teams. The objective of this research paper is to explore onboarding processes of newly graduated software developers joining remote agile teams in large Norwegian IT-consultant companies. This will include looking at used onboarding strategies, agile practices and activities and the success of it all. The theoretical foundation includes an introduction to onboarding, remote work and agile software development, as well as related work to these topics. Additionally, a presentation of Ju et al.'s (2021) three onboarding strategies will be made.

The results in this master thesis are based on thirteen interviews collected in a multi-case study with a total of six cases. The main findings of this study show that some onboarding strategies are more suited for remotely onboarding newly graduated developers than others and that the most important thing regarding onboarding strategies is the value of having a plan. When it comes to the use of agile practices and activities, they can support the onboarding in a good way, especially collaborative activities like mentorship or pair programming. The final takeaway from this report is that similar onboarding experiences can vary depending on the onboarder, and therefore, each onboarding process should be adapted to the given scenario.

Keywords: onboarding, remote work, agile software development, newly graduated software developer

Sammendrag

Motivasjonen bak denne studien stammer fra den nylige økningen i fjernarbeid blant programvareutviklingsteam, behovet for å gi en nyutdannede systemutviklere en varm velkomst til teamet og viktigheten av høykvalitetskommunikasjon i smidige team. Målet med studien er å utforske onboardingprosesser hvor nyutdannede systemutviklere, ansatt i store norske IT-konsulentselskaper, slutter seg til allerede eksisterende smidige utviklingsteam. Dette inkluderer å se på hvilke onboardingstrategier og smidige praksiser og aktiviteter som har blitt brukt, samt suksessen av disse. Det teoretiske grunnlaget inkluderer introduksjoner og relatert arbeid til onboarding, fjernarbeid og smidig utvikling. I tillegg vil Ju et al.s (2021) tre onboardingstrategier bli presentert.

Resultatene i denne masteroppgaven baserer seg på tretten intervjuer innsamlet i en multi-casestudie med totalt seks caser. De viktigste funnene viser at noen onboardingstrategier er bedre egnet for å onboarde nyutdannede utviklere enn andre og at det er stor verdi i å ha en god og gjennomtenkt plan for onboardingstrategiene. Når det kommer til bruken av smidige praksiser og aktiviteter, kan de være med på å støtte onboardingsprosessen på en positiv måte, spesielt med tanke på samarbeidsaktiviteter som mentorordninger eller parprogrammering. Den endelige konklusjonen fra denne rapporten er at onboardingprosesser kan variere avhengig av den som blir onboardet, og bør derfor tilpasses den gitte situasjon.

Nøkkelord: onboarding, fjernarbeid, smidig utvikling, nyutdannet systemutvikler

Preface

With great happiness and gratitude, I present my master's thesis in Computer Science for the *Department of Computer Science* (IDI) at the *Norwegian University of Science and Technology* (NTNU), Trondheim. This is a final assignment in my education, and it marks an important milestone in my academic career.

Throughout my time as a student, I have taken a broad selection of courses within the Computer Science field. However, agile software development and teamwork have always caught my interest the most. In 2021, when I wrote my bachelor's thesis, I explored the impact of Covid-19 on the productivity of agile teams. I knew that this was a field I wanted to delve deeper into in the future.

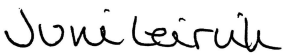
My inspiration for this thesis stems from the experiences of friends and fellow students who faced unsatisfactory onboarding processes during lockdowns caused by Covid-19. I wanted to look into this to hopefully discover some triggering events so others might have better experiences in the future. Additionally, as I will be going through my first onboarding process in a couple of months, it further motivated my decision to explore this topic.

Acknowledgments

First of all, I would like to express my genuine gratitude to my supervisor, Torgeir Dingsøyr, for his invaluable guidance and support throughout the semester. His expertise and encouragement have been instrumental in shaping this work. I also want to express my gratitude to the other students in my mentoring group. They have motivated me throughout this project, and they have been willing to offer advice that has helped to improve my comprehension of the subject significantly. Our combined efforts have created a highly rewarding environment for collaborative learning.

Furthermore, I am deeply grateful to my friends at *Gamle Fysikk*, my study hall. They have provided their kind words and meaningful conversations during moments of lowering motivation. Their presence and encouragement have been a constant source of inspiration, reminding me of the importance of perseverance and fellowship.

Lastly, I would like to thank my friends and family for their great support throughout my entire education. Without the support of the individuals mentioned, the completion of this project would not have been possible. I am truly fortunate to have had such exceptional guidance, encouragement, and friendship throughout this journey.



Juni Leirvik Larsen

Trondheim, 03.06.2023

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background & Motivation

Starting a new job or joining a new team can be exciting but stressful for most people. This applies especially if it is your first-ever professional job. There are a lot of new people, new surroundings, and a lot to learn. Receiving a warm welcome from the new company and the new team will be highly appreciated in order to make the transition to the new everyday work life as smooth as possible. Therefore, successful onboarding is often necessary, but it might be extra crucial for recent graduates (Pham et al., 2017).

Onboarding is a process where the goal for new additions to the team is to move from being team outsiders to becoming team insiders (Bauer and Erdogan, 2011). To accomplish this, the new members should get an understanding of their job responsibilities, the culture context, standard of work, and development process and gain project knowledge (Buchan et al., 2019). DeMarco and Lister (2013) stated in their book that *"We all know that a new employee is quite useless on day one or even worse than useless since someone else's time is required to begin bringing the new person up to speed"*. If the onboarding experience is poorly executed, it can cause the new team member to develop anxiety by its lack of team contribution and trust, as well as a reduction in overall team productivity (Buchan et al., 2019).

The onboarding process is usually extremely expensive for a team (Rodeghero et al., 2021), and its expenses can depend on various factors like how time-consuming the process is and the frequency of onboarding (Buchan et al., 2019). How time-consuming it is can depend on the new developer's previous work experience, skill

level, and attitude to learning and adapting to the team (Buchan et al., 2019). Newly graduated software developers with little work experience will often require more resources in the onboarding process than more experienced developers. The complexity and the expenses of onboarding show the importance of streamlining as much as possible without compromising its quality. This also applies when working in a remote team. However, when working remotely, a different approach might be necessary due to people not being able to communicate and socialise in the same way as when working in the office.

Working *remote* is commonly used to describe a developer that works away from the physical office. The idea of remote work is that it is possible to work in any chosen location and effectively contribute to a development team (Rodeghero et al., 2021). Working remotely is not something new, but in March 2020, there was an unexpected increase in employees working from home when something that almost no one ever thought would happen. A global pandemic caused by a highly contagious virus called Covid-19 caused country by country to shut down, and suddenly, it became a matter of staying home and having as few close contacts as possible to prevent people from getting ill. The whole world, including schools and workplaces, was affected in one way or another. The traditional workplace changed overnight, and most employees, including members of agile software development teams, had to transition to remote work and start working from home. This did not only change the usual way people work together and how developers collaborate but also how the onboarding process unfolded (Rodeghero et al., 2021). Even though remote work had an upswing when the Covid-19 pandemic started, working remotely is not new. In 2020, Micaela published a report with statistics from Eurostat showing that more than 40% of workers in IT and other communication services already were working from home regularly or at least with some frequency in 2018 within the European

Union (EU). This helps to show that even though the pandemic is now coming to an end, many of these changes will remain in the future, and remote work is here to stay.

Working remotely can be especially challenging in agile software development teams. The agile methodologies are iterative and incremental approaches to software development that aims to increase the frequency of deliveries, embrace higher rates of change by working closer together and have more frequent communication (Beck et al., 2001). Since agile emphasizes close collaboration, effective communication, and shared ownership, working remotely can lead to some challenges. Remote work adds another layer of complexity to the agile methodology, as team members are physically separated and rely heavily on digital communication tools. In the case of remote teams, a well-executed onboarding process can help to ensure that new team members understand the principles, practices, and values of agile, as well as the specific team dynamics (Gregory et al., 2022).

1.2 Scope & Research Question

The scope of this study is to collect and discuss data regarding experiences with remote onboarding of newly graduated software developers to agile software development teams. Based on the need for a warm welcome to the team, the recent increase in employees working from home, and the need for high-quality communication in remote agile software development, the following research question has been chosen:

RQ: How are newly graduated software developers onboarded to already existing remote agile software development teams in large Norwegian IT consultant companies after the outbreak of the Covid-19 pandemic?

This research question will include exploring experiences with different onboarding strategies and how agile principles and activities contributes to an onboarding process. It will also be discussed whether the onboarding experiences of the different informants were successful or not based on the challenges and highlights they experienced and their personal opinions. In this process, the agile methodologies, scrum and extreme programming, and onboarding strategies found by Ju et al. (2021) will be used. Ju et al. conducted a study where 32 developers and 15 managers were interviewed and uncovered three onboarding strategies that managers commonly use to onboard new developers. This study will use the same three strategies and compare their use of them with the success of remote onboarding of newly graduated developers. It is important to be able to present the best possible onboarding experiences. It can at a later stage contribute to discovering what further research should look into and hopefully improving remote onboarding experiences for newly graduated software developers in the future.

As the research question mentions, the scope includes looking into large Norwegian IT consultant companies (Eurostat, n.d.). This implies that the companies of everyone involved in this study have the same starting point for arranging an onboarding. Large Norwegian IT consultant companies should all have the finances and expertise to onboard developers in a satisfactory way. Researching consulting companies is intriguing due to their frequent team turnovers and the fact that they are often the starting point for newly graduated students entering the professional workforce. The scope also includes newly graduated software developers and their first-ever professional agile software development team. As mentioned in subsection 1.1, this is because of the extra need for follow-up when it is your first-ever professional job. It is important to note that the developers being studied have been hired by external customers of the consultant companies for specific projects.

Since there has been some time since the Covid-19 pandemic first started, many people have experienced onboarding into a hybrid or remote team. Numbers have also recently been published showing that working remotely is still very common, even after most parts of the world have reopened after the pandemic (Barrero et al., 2023). This indicates that remote work is here to stay, and therefore, this is an appropriate time to do the research, even though the pandemic seems to be over for now in most parts of the world.

1.3 Scope Limitations

This research project will result in a multi-case study where the selected topic is presented, researched and discussed. However, the scope of this project has some limitations. This report is a master's thesis given by the *Norwegian University of Science and Technology* (NTNU) worth 30 points in the *European Credit Transfer and Accumulation System* (ECTS). This study is also carried out within a set time period of 20 weeks.

As mentioned, the scope of this study focuses on newly graduated developers hired by large Norwegian IT-consulting companies and working in remote agile software development teams. Despite this specific scope, there are numerous potential sources of informative data available. However, due to the time constraints of this study, only a limited amount of available data sources have been included, which means that some relevant information may not have been incorporated.

The theoretical foundation of this study is based on previously published research on the topics. Because of the extent of this thesis and the amount of research published on this topic, it has not been possible to review all available material on the topic.

The material is limited to what has been found interesting and relevant to the author. There are also several categorizations of onboarding strategies available. In this study, it has been chosen to focus on one of them and use it as the basis for the study. Therefore, the theoretical part of this study is not a summary of all available research on the chosen topic. It is a selection of available resources that were possible to review in the given time period.

This study uses informants that have been onboarded remotely after the outbreak of the Covid-19 pandemic. During this time, the pandemic and society, in general, have been through many different phases. This means that the results from the different informants will depend on when they onboarded the team and for what reason the team worked remotely. An attempt has been made to take this into consideration. In summary, what has been mentioned in this chapter, may result in some important information and facts being missed which could be relevant.

1.4 Target Audience

The main target audience for this report includes researchers and students who have an interest and background in the current topic. Additionally, since this report focuses on onboarding processes in large IT consultant companies, it is also relevant for recruiters and software developers. The study primarily focuses on the field of software engineering; however, certain findings may have relevance and applicability to other fields of education. Reading this report requires a basic understanding of computer science, agile software development, and onboarding processes. Readers without this experience are strongly encouraged to refer to the background literature and theory provided in section 2 for a better understanding of the research. Section 2 provides

an explanation of the necessary knowledge required to comprehend the content of this research. Furthermore, additional knowledge about the impact of Covid-19 on the world and an understanding of the typical workday of an agile software developer in an IT consultant company may be beneficial.

1.5 Report Structure

Section 1 - Introduction

This section, section 1, has included the background and motivation for choosing to do this research. It has also presented the chosen research question, the scope and its limitations, and the target audience for this research project.

Section 2 - Background Literature & Theory

The next section, section 2, aims to present and explain the necessary principles and theory to be able to understand the rest of the report. Since, as mentioned in subsection 1.4, some background knowledge is expected, only key elements and theories are presented.

Section 3 - Method

Section 3, the method section, presents the research method used and how the research was conducted. This includes a description of the case, the selection of the participants, the interviews conducted and the analysis of the data. This section concludes with an evaluation of the method, and its limitations are presented.

Section 4 - Results

The results are presented in section 4. This section includes the analysis of the data from the research and a presentation of them. These results are aimed directly

towards the research question.

Section 5 - Discussion

The next section, section 5, discusses the results from section 4 and uses theory from section 2. The last part of this section provides an assessment of the limitations of the study.

Section 6 - Conclusion

Lastly, section 6 presents a conclusion to the research question based on the results and discussion from the previous sections. At the end of the section are suggestions for future work.

2 Background Literature & Theory

This section will serve as a theoretical foundation in order to discuss the results later in section 5. The readers will also be presented with relevant information in existing literature and theory.

The main focus of this section will therefore be the purposes of the research question; studying onboarding strategies, looking into the use of agile methodologies in onboarding and the success of those. Firstly, in subsection 2.1, general information about onboarding, including onboarding success and remote onboarding, is presented. Secondly, in subsection 2.2, there will be given an introduction to Ju et al.'s (2021) three onboarding strategies. This includes related work to these strategies. Finally, in subsection 2.3, an introduction to the basics of agile software development, some common agile methodologies and how to onboard an agile team.

2.1 Onboarding

The onboarding process can be defined as *"a procedure whereby new employees move from being team outsiders to becoming team insiders"* (Bauer and Erdogan, 2011). Onboarding is an essential process at both organizational and team levels in the IT industry (Gregory et al., 2022). It is the direct bridge between the promise of new employee talent and the attainment of actual productivity (Snell, 2006). When working in a software development team, it is normal for new members to occasionally join the team to replace a leaving member or augment the team's capacity. New members must be integrated properly to ensure that they become productive and trusted contributors to the team (Buchan et al., 2019). This can typically consist of

working with people, undertaking activities or working with artefacts (Buchan et al., 2019). The developers that onboard a team will be referred to as *onboarders* in this report.

When studying to become a software developer, universities strive to prepare students for the professional industry to the best ability. However, Begel and Simon (2008) from Microsoft Research has experienced that newly graduated software developers become novices all over again as soon as they enter the professional software engineering workforce. It is normal for existing members of agile teams to recognize that students entering the workforce directly from university often do not have the complete set of software development skills needed to be productive, especially in large, independent software development companies. It has been reported that newly graduated software developers lack communication and teamwork skills, are unprepared for complex development processes, legacy code, and deadlines, and fail to work with limited resources (Taft, 2007). These are all skills that should be focused on in the onboarding process.

2.1.1 Onboarding Success

Onboarding is a critical process for any organization or team to ensure the successful integration of new employees. To determine what is required for an onboarding process to be successful, can however be challenging. In a study by Bauer and Erdogan (2011), an attempt to do so was made. Four building blocks, known as the Four C's, have been identified as necessary for successful onboarding. According to this study, these building blocks define the success of onboarding. These are mainly used for onboarding on an organizational level, but the principles can also apply on a team

level. The goal of an onboarding process is to achieve all four building blocks, as they provide a stable foundation for new employees.

Connection
Culture
Clarification
Compliance

Table 1: Building blocks of successful onboarding (Bauer and Erdogan, 2011)

1. **Compliance** is to teach employees basic legal and policy-related rules and regulations.
2. **Clarification** is to ensure that newcomers understand their new jobs and related expectations.
3. **Culture** is to provide newcomers with a sense of organizational norms.
4. **Connection** is to the interpersonal relationships and information networks that newcomers must establish.

The success of the onboarding process is not guaranteed. If poorly managed, onboarding can be inefficient and costly for the organization or team (Snell, 2006), leaving onboarders with little to gain. A well-designed and automated onboarding process can reduce costs, hasten the time to productivity, and improve employee satisfaction and retention (Snell, 2006).

Depending on the onboarder, onboarding success may also vary. Moe et al. (2020) discovered that the outcome of an onboarding process could be different even when an

organization or team applies the same activities and strategies for all the onboarders. This is because onboarding can be affected by several factors, including the project's complexity, team types, and how busy the team are. They also state that the actual outcome of the onboarding is hard to predict in advance. Therefore, Buchan et al. (2019) suggests that onboarding processes should be adapted to each given scenario and onboarder throughout the process to get the best possible experience.

When it comes to onboarding duration, teams have reported that they expect to support the onboarding of a new team member for around three months (Buchan et al., 2019). Ju et al. (2021) wrote that others expect it to take over six months.

2.1.2 Remote Onboarding

Onboarding is also needed in remote software development teams. Working remotely can be done in many ways and for many reasons. Some developers work full-time remotely, while others only work a couple of days a week away from the team. Working part-time remotely or having parts for the team working remotely is normally called *hybrid work* (Deshpande et al., 2016). Another way of working remotely is *distributed work*. Distributed work is when the team work together across different locations and times (Deshpande et al., 2016). Different types of remote work will affect teams differently, but the goal and purpose still remain the same in all software development teams. Even though remote work has been around before the Covid-19 pandemic, it is important to note that remote work during a pandemic is not the same as traditional remote work.

Both Moe et al. (2020) and Rodeghero et al. (2021) have studied onboarding in different remote teams. The goals were to understand the remote onboarding process

and the challenges that onboarders faced during the process. These studies discovered challenges including finding documentation, missing domain knowledge, communicating, asking for help, unclear tasks and bonding with teammates (Moe et al., 2020; Rodeghero et al., 2021). Recommendations like creating a customized plan based on the onboarders' needs and providing onboarders with both an onboarding mentor and a technical mentor were mentioned as the best measures in a remote onboarding process (Moe et al., 2020; Rodeghero et al., 2021).

Moe et al. (2020) study looked into mentoring during onboarding. However, one of the examined teams had limited success because the mentor had no time to do their job. The study also concluded that mentoring is a challenging job, and the mentors themselves need help in performing that job and balancing mentoring and solving their own tasks. However, the study was still positive to the mentor arrangements in remote teams.

2.2 Onboarding Strategies

This study will use three onboarding strategies that, according to Ju et al. (2021), represent the majority of onboarding strategies when assigning tasks in practice. They concluded, after conducting interviews with 32 developers and 15 managers in Microsoft, that *Simple-Complex*, *Priority-First*, and *Exploration-Based* were the best representation of the used onboarding strategies. These strategies based on Ju et al.'s (2021) study are presented below.

2.2.1 Simple-Complex

In this strategy, the complexity of the assigned tasks of the newly hired will gradually increase with time. The task complexity will be consistent with the leaders' expectations when onboarding new team members. In the beginning, the tasks will have low risk and be technical unchallenging but provide experience with tools, processes, technology and team norms (Buchan et al., 2019). Managers use this method to help developers achieve expected learning goals. Using this method helps new developers to build high confidence fast.

In a study on remote onboarding by Rodeghero et al. (2021), a recommendation to assign simple tasks first was made. They found out that onboarders often take a few weeks to create their first pull request. By completing a simple task, such as fixing a spelling error, onboarders are allowed to quickly go through the process of building a project, creating a pull request, submitting it, and having it reviewed. Without completing this process, they may not understand the workflow as quickly as they should and, consequently, not understand the discussions during team meetings until after they have gone through the process of creating their first pull request.

Some of Buchan et al.'s (2019) interviewees described the value in undertaking the simple task as being the opportunity to apply and contextualize their recent learning. The importance of the low impact of making a mistake in a simple task or the low impact of being slow to finish a simple task was noted. As opposed, in the same study, a more experienced developer found being "thrown at the deep-end" on a complex task useful during onboarding because it was an opportunity to prove themselves to the team as a step towards gaining their trust and respect.

Gregory et al. (2022) found that giving inexperienced onboarders small and simple

tasks are not a contradiction to working agile. It was always possible for the team to give onboarders smaller tasks because they always had a set of reasonably small tasks on the backlog.

2.2.2 Priority-First

This method is mainly used by teams under pressure, and it allows developers to generate value for the team immediately. The onboarders will get assigned tasks following the backlog's order. This approach is based on the idea of prioritizing the tasks most essential to the team's success, will help onboarders get up to speed and start making meaningful contributions quickly.

This onboarding strategy can, according to Ju et al. (2021), have various degrees of success. One interviewee reported he was highly motivated because his coworkers were admiringly focused. He felt safe even if he made mistakes or asked for help because the team was too busy to worry about little things like people being disturbed or offended. On the other side, high-priority tasks can be challenging. "Challenging tasks early in the onboarding process could spur fears and low confidence" (Ju et al., 2021). However, having good team support can counter some negative influences.

Gregory et al. (2022) carried out a study about how onboarders integrate into an established agile project team. This study found that a team that uses agile practices and the Priority-First onboarding strategy can support the onboarding process in a good way. By starting to use agile practices from day one and giving the onboarder tasks from the backlog, the onboarder will start doing productive work immediately. In addition, agile practices will be learned from the beginning. After two weeks, onboarders will have gone through a whole sprint and be familiar with several agile

practices and the major parts of the team's process. According to this study, this makes the Priority-First onboarding strategy a good choice in agile teams.

2.2.3 Exploration-Based

In this strategy, the onboarders are assigned loosely defined tasks. The assigned tasks are normally outside the team's core production, so onboarders can explore without time pressure or worrying about breaking the production code. This is a strategy normally used to onboard senior developers or used in newly established teams. In newly established teams, it is an opportunity to explore and learn together as a team since the team has many unspecified tasks that need to be done. Nevertheless, it is important for the onboarding managers to make it clear that the tasks are meant for the onboarders to learn. Such a strategy can have negative effects on newly graduated developers considering that they might feel pressure by not delivering as frequently as the more experienced developers. However, Ju et al. (2021) reports that newcomers feel more comfortable with this strategy in new teams than other teams.

2.3 Agile Software Development

Onboarding into an agile team may differ from onboarding to other types of teams (Britto et al., 2018; Gregory et al., 2022). Before presenting some related studies to onboarding an agile team, a brief introduction to agile software development will be given.

As mentioned in the introduction, agile software development is an iterative and incremental software development method based on the Agile Manifesto. This means

that systems are developed through repeated cycles and in smaller portions at a time. The Agile Manifesto was created in 2001 when seventeen developers came together due to a shared frustration (Beck et al., 2001). The frustration was about the current state of affairs where companies were so focused on excessively planning and documenting their software development cycles that they lost sight of what really mattered - pleasing their customers (Beck et al., 2001). The seventeen representatives had backgrounds from different documentation-driven, heavyweight software development processes (Beck et al., 2001). This resulted in the formation of the Agile Manifesto. It consists of four values (See Figure 2) and twelve principles, where the main goal is to focus on the people and not the processes and tools.

Individuals and interactions	over	processes and tools
Working software	over	comprehensive documentation
Customer collaboration	over	contract negotiation
Responding to change	over	following a plan

Table 2: The four values of the Agile Manifesto (Beck et al., 2001)

There is a need for team members to work together, communicate and collaborate to ensure a successful agile software development project (Meier et al., 2016). Kraut and Streeter (1995) has defined communication in software development as different people working on a common project agreeing to a common definition of what they are building, sharing information and meshing their activities. And collaboration can be defined as brainstorming and working together as a team (Rodeghero et al., 2021). Frequent communication is one of the best ways to build trust in a software development team and make the development more efficient (Pikkarainen et al., 2008). Wu et al. (2008) has reported a relationship between face-to-face communication and productivity in agile development. In a traditional work environment, the members of

agile teams are seated together to stimulate and improve informal and open communication, which increases the possibility of face-to-face communication. The frequency and formalization of the information exchange are being challenged as soon as members of agile teams start to work from separate locations. However, communication is still important for remote software development (Kraut and Streeter, 1995).

Scrum

Several different development methodologies are based on the principles of the Agile Manifesto. The most common agile methodology is Scrum. Scrum is an iterative approach divided into iterations, cycles or *"Sprints"*, which last over two to four weeks (Sommerville, 2011). Each sprint aims to create and develop one part of the software based on prioritized requirements and user stories. The *product backlog* is the starting point for planning a sprint, and it is a list of things that need to be done to complete the project (Sommerville, 2011). During the *sprint planning*, the project team works together with the customer in order to select the features and functionality to be developed during the sprint and to include these in a *sprint backlog* (Sommerville, 2011). After this, the development starts. Every day the developers meet for a *daily stand-up meeting* where the goal is to let team members inform each other on the progress, their current work status and if any help is needed to continue the work. When the sprint is completed, the team performs a *sprint review* and a *retrospective*. This includes discussing what each of the team members did during the sprint and the process of evaluating for future improvements and learning for the project and the team. A visualisation of the entire scrum process is presented below in Figure 1.

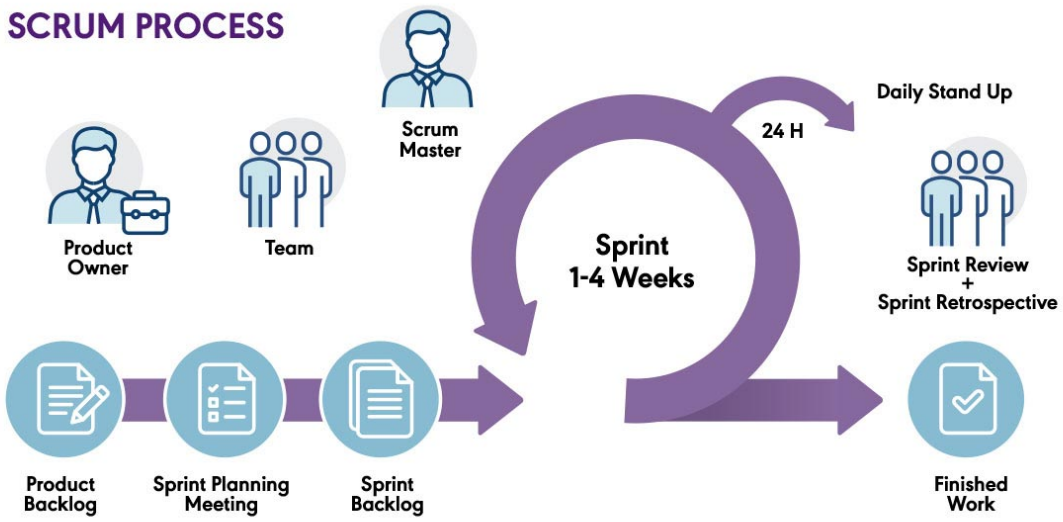


Figure 1: The Scrum Process (Pmadmin, 2023)

Extreme Programming (XP)

Extreme Programming (XP) is another agile methodology. While Scrum is a framework to help teams develop complex projects in an adaptive manner and not a dictate on how to do the work, XP puts instead much emphasis on good programming practices (Beck, 1999). Having small and constant releases is also an important part of XP. One of the key values in XP is *feedback* (Bell, 2017). XP embraces change and strives to receive early, constant feedback. Feedback comes in many shapes and sizes. When *pair programming* or *mob programming*, the comments of your peers are vital feedback. Pair programming means that two programmers work together at one workstation, while mop programming includes even more developers (Buchan et al., 2019). *Tests* are another source of precious feedback that goes beyond the test results. To determine whether writing code is easy or hard is also a kind of feedback.

If writing the tests is hard, the design is probably too complex (NimbleWork, 2023). Therefore, implementing features using the test-first approach is a major part of XP. Customer involvement and using code review are other important forms of feedback (NimbleWork, 2023). Code review is "a manual inspection of source code by developers other than the author, is recognized as a valuable tool for reducing software defects and improving the quality of software projects"(Bacchelli and Bird, 2013). In XP, customers work closely with the development team, providing constant feedback on requirements, priorities, and functionality. This direct interaction helps to clarify expectations, validate assumptions, and ensure that the software aligns with customer needs. Other key values in XP are communication, respect, simplicity and courage (Bell, 2017).

As well as with other software development teams, there has been an increase in the number of remote agile teams these last few years. Marek et al. (2021) mentioned that agile software development teams were able to transform to remote work without much turbulence when the Covid-19 pandemic started due to the popularity of distributed and remote agile software development teams prior to the outbreak.

When working with software development remotely, some of the most commonly reported successes were stand-up meetings, sprints, continuous integration, sprint planning, retrospectives, pair programming, and sprint review (Deshpande et al., 2016). These are all typically used activities when using agile methodologies. By using enforced communication practices of agile development, some of the challenges for remote teams can reduce temporal, geographical, and social-cultural distances (Deshpande et al., 2016). However, it was discovered that the majority of distributed remote teams modified the use of agile practices because of situational requirements (Deshpande et al., 2016).

2.3.1 Onboarding an Agile Team

Onboarding into an agile team may differ from onboarding to other types of teams. Gregory et al. (2022) found that a team that uses agile practices can support onboarding. One way to do this is by starting to use agile practices from day one and doing productive work immediately. For instance, by attending daily stand-ups, onboarders learn the purpose and the structure of those meetings. It is also a very effective way of transferring knowledge about the product requirements, the technology being used, coding styles and conventions, and the people in the team.

In a study conducted by Britto et al. (2018), three globally distributed agile software teams were studied. One of the key findings was that onboarding remote developers to an ongoing agile project was the greatest challenge. It may be hard for developers to start being productive when onboarded remotely. The reason for this finding was that the agile approach involved minimal documentation, meaning that new developers had to engage in continuous dialogue with mentors to understand the project. Other interesting findings in this study were the importance of explaining the expectations, adapting the onboarding strategy to the onboarder and providing feedback. Lastly, it was mentioned that the team's company should invest in travels for the onboarders when working remotely. In one of the cases, it was stated that it is very important to travel to other sites in order to meet people face-to-face. This way, it will be much easier to ask for help and to be aware of what actually happens. It's almost like going to another room in your workplace, except from the distance.

2.3.2 Onboarding Techniques

Although the objective of every onboarding procedure is the same — to have the onboarder become a productive member of the team as quickly as possible — there are various techniques that can be used to accomplish this (Buchan et al., 2019). In this study, onboarding techniques refer to activities and experiences designed to enhance the onboarding process and increase the likelihood of achieving desired outcomes for the onboarder. Buchan et al. (2019) provides a list of five onboarding techniques to be considered when onboarding a developer to a remote agile team.

1. *Socialization opportunities* is first on the list. The study considered involvement in social events in the team as a significant factor in developing relationships and team trust. Team socializing is really important, especially from an onboarder's perspective.

2. Next is having *access to high-quality knowledge artifacts.* An effective and efficient onboarding relies on the onboarder receiving timely and low-effort access to relevant information, as well as the quality of the information. Another researcher recommends that new developers should be able to write code on their own, relying on pull requests and code reviews to provide feedback rather than having explicit sessions to describe code or architectural conventions (Viviani and Murphy, 2019).

3. Third on the list is *access to formal training.* The team may need to organize for the onboarder to attend training courses to meet their anticipated and discovered skill gaps.

4. *Proactive feedback and knowledge sharing* is the next technique on the list. This includes mentoring, peer support, use of online communities and digital

communication. Having a mentor is probably one of the most used and recommended onboarding techniques, and it is extremely valuable for successful new employee onboarding (Bauer and Erdogan, 2011; Buchan et al., 2019; Gregory et al., 2022; Viviani and Murphy, 2019). A mentor is normally a more experienced team member or colleague that teaches the onboarder about the project, offers advice, helps with job instruction, and provides support in social and political terms (Bauer and Erdogan, 2011).

5. The final item on the list is to *provide psychological safety*. The onboarding is more effective if the onboarders feel that they will not be punished or blamed for making a mistake or being less productive than other team members in the team to produce work.

The results of this study clearly showed that mentoring was recognized as an important onboarding practice providing information and advice to the onboarder, as well as acting as a confidant.

3 Method

In order to address the research question presented in subsection 1.2, this section outlines the explanation for choosing to do a case study, the chosen cases, and an overview of the overall research process. It also provides insights into the participants involved in the research and the reasons for their selection. Finally, an evaluation of the chosen research method and its limitations is provided.

3.1 Research Method

The chosen method in this research is a qualitative exploratory multi-case study. Due to the chosen research field and a desire to look at individual developers' experiences, empirical research is chosen. Empirical research may be quantitative or qualitative, but a qualitative study provides a richer and deeper description (Runeson and Höst, 2019). A case study is typical qualitative research, and it was selected to be able to investigate the relations, how those are connected and how they are affecting the case (Oates et al., 2022). Case studies are typical flexible design studies because the key parameters of the study may be changed during the course of the study (Runeson and Höst, 2019). By exploring cases and painting a detailed picture of how they link together, the researcher will try to explain how and why certain outcomes occur in different situations. An exploratory case study is finding out what is happening by seeking new insights and generating ideas and hypotheses for new research (Runeson and Höst, 2019). This is efficient in this research since an exploratory study often is used when literature has limited resources, and a real-life investigation is suitable to collect more data (Oates et al., 2022).

Methodology	Primary Objective	Primary Data	Design
Case study	Exploratory	Qualitative	Flexible

Table 3: Research methodology characteristics (Runeson and Höst, 2019)

Runeson and Höst (2019) suggest that there are five major process steps to be walked through when conducting a case study. These steps, which are presented in Figure 2, summarise the overall process that was adopted for this study.

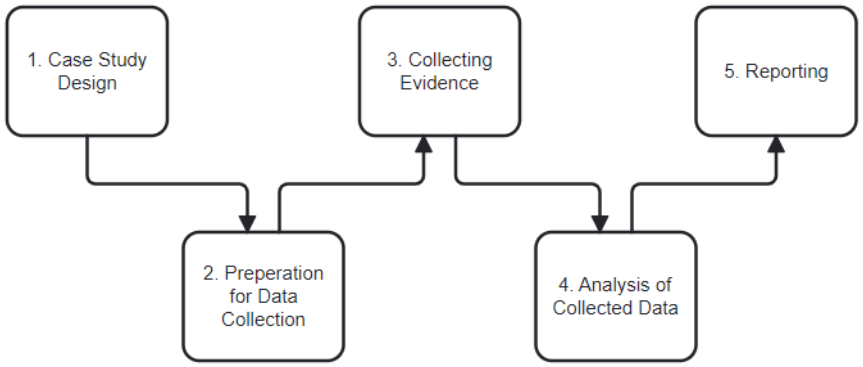


Figure 2: Main steps in case study research (Runeson and Höst, 2019)

The first step in Runeson and Höst's (2019) guideline for case studies is *case study design*. This includes creating a plan for the case study research by defining objectives, case and research questions, as well as deciding how and where to collect data. *"...good planning for a case study is crucial for its success"* (Runeson and Höst, 2019). How this was carried out in this research is presented in subsection 1.2 and 3.2. The next step, *preparation for data collection*, includes deciding on the data source and interview questions and obtaining necessary approvals. *Collecting evidence* is the third step. This step consists of conducting interviews, making audio recordings, writing notes and making sure that the right data is collected and that it is done properly. More details on steps two and three are presented in subsection 3.3. The

fourth step is *analysis of collected data*, and how the data was analysed and which tools were used is presented in subsection 3.4. The last step in the guideline is *reporting*. This is the process of writing this research paper.

3.2 Case Study Design

As mentioned, for this case study, an *exploratory* objective was chosen. This includes *"finding out what is happening, seeking new insights and generating ideas and hypotheses for new research"* (Runeson and Höst, 2019). In exploratory case studies, the collection of evidence may be undertaken prior to the definition of the research question (Tellis, 1997), and this also applies to this research. A proposal for a research question was made at the beginning of this research, but the final one was defined during the analysis of collected data. It was also decided to have a multi-case design in this research. This approach was selected to provide a broader range and different aspects to the analysis, which can be used to discuss similarities and differences in the cases. The results and evidence of multiple case designs are more convincing, which makes this design more robust compared with a single case study design (Yin, 2009). The defined cases are presented below.

3.2.1 Cases

The chosen cases in this study are all different remote agile development teams that have onboarded a newly graduated software developer from large Norwegian IT-consultant companies after the outbreak of the pandemic. A total of six cases have been researched in this report.

Case 1

A complete scrum team consisting of eight members working remotely due to long commutes. The team consist of four developers, a UX designer, a tester, a product owner and a scrum master. The team has been working remotely since the start of the project, which has been ongoing for approximately six months. Although the team primarily operates remotely, they gather at the office once a week for collaborative work. This team's project was creating an application for an external private organisation. This particular case was chosen because of its highly engaged onboarding leader and the team's weekly office meetings.

Case 2

The team, consisting of five members, including the onboarding developers, was forced to transition to remote work due to Covid-19 restrictions. They swiftly adapted and transformed into an agile team when the pandemic began. The project had been ongoing for respectively six and eight months when the developers were onboarded. The project concerns increasing the security of an existing system at a public organisation. Prior to this study, one of the team members had highlighted possibilities for improvement in remote onboarding within this team. Therefore, this case presented an intriguing opportunity for further investigation and analysis.

Case 3

An agile distributed team consisting of five developers and one project leader working from several various locations in Norway. The first remote onboarding took place after the project had been ongoing for nine months, and the subsequent onboarding occurred around a year. This team has never met in person, which makes this a relevant team to research. The team is currently engaged in developing a new phone application for a small start-up company.

Case 4

An agile team using principles from XP consisting of ten team members, mostly developers, creating a system regarding map handling for a public organisation in Norway. The team members, in this case, worked remotely because of a long commute. One day a week, the team comes together and works physically together from the office. When the onboarding happened, the project had already lasted for one and a half years.

Case 5

The team operates remotely as a result of the Covid-19 restrictions. They have adopted a modified scrum approach tailored to their needs. This team have onboarded onboarders every year in the last couple of years. At the time of this research, this team had been ongoing for more than three years. This team consists of between 15 and 17 team members, mostly developers. The project consists of creating a complaints service for a large public organisation. Considering the project's extended duration and the team's regular onboarding procedures, this case held significant relevance for the study.

Case 6

An agile team consisting of a total of six members working remotely due to a lockdown. This team consist of four developers, one project leader and one designer. At the time of this research, the project has been ongoing for one year. This team had to transfer to become a remote team when the Covid-19 pandemic started. The researcher chose this case due to the sudden change to remote work.

3.3 Data Collection

In preparation for data collection, the data source and interview guide were decided, and necessary approvals were obtained. One of the necessary approvals was from Sikt (Norwegian Agency for Shared Services in Education and Research) to be able to process personal data in this research project. Another necessary approval was acceptance from the informants that it was OK that they participate in this study. This approval guide is added in Appendix A. The chosen data source was a direct method, which means contacting the subjects directly and collecting data in real-time (Runeson and Höst, 2019). An advantage of this data source is that the researcher can, to a large extent, control exactly what data is collected, how it is collected, in what form the data is collected, which the context is etc. (Runeson and Höst, 2019). In the beginning, the researcher's contacts were reached out to. Further, the snowball method was used.

3.3.1 Selecting Participants

Three different types of roles have been interviewed in this study. This is because it helps to get a better insight into how the onboarding processes of cases actually unfold and to avoid personal opinions as much as possible. The focus of this study is still the developers that onboard the teams, and participants from this role will therefore be the majority. The other roles will only provide information that can supplement what the onboarder says.

There were some criteria for the participants, depending on their role in this study. The participant had to fall into one of the following categories:

The participant ...

1. ... has onboarded a remote agile team as their first-ever full-time software development team (Onboarder).
2. ... was part of a remote agile team when the onboarding process of a newly graduated software developer was happening (Existing team member)
3. ... has been in charge of onboarding a newly graduated software developer to a remote team (Onboarding leader)

In addition to these criteria, the participant had to have time and agreed to participate in the interview and this study.

3.3.2 Semi-Structured Interview

Semi-structured interviews were used to collect evidence in this study. The reason for this choice was to allow for improvisation and exploration of the studied objects. It also allowed for the inclusion of additional questions if the informants raised issues not covered by the prepared questions (Oates et al., 2022). This approach was beneficial as the researcher did not have much experience with remote onboarding before, and therefore, the chance of something unexpected and new being mentioned was high. While most interview questions were planned, some modifications were made during the interview process, and the order of questions occasionally changed based on the informant's responses and the flow of the conversation. The interview questions also varied depending on the different roles.

Although the interviews could vary, they had the same basic structure. This structure is inspired by Runeson and Höst (2019), and it is shown below.

1. Introduction
2. Fundamental questions
3. Main part
4. Ending

The first part of the interview, *introduction*, consisted of a presentation about the researcher, the research project, and the privacy agreement. The *fundamental questions* included basic questions about the informant's background and project. The *main part* of the interview is the most extensive part. This includes most of the research questions as well as is an opportunity for the informant to talk freely regarding their first-ever onboarding experience. If the informant is not an onboarder, these questions will naturally change to ask the questions in a more correct way. The *ending* consists of summarizing, thanking and closing the interview. The complete interview guide for this study is added in Appendix B.

Before the main interview process started, a pilot interview was conducted. To get the best possible feedback from the pilot interview, the chosen informant passed all criteria for an onboarder to participate in this research. During the interview, the informant was asked to say the interpretations of what was being asked. After the pilot interview, some parts of the introduction and the main part were reformulated or removed. Some of the feedback received was to have a more detailed presentation in the beginning and be careful with leading the questions in a direction. The answers from the pilot interview are not a part of the results of this study.

In total, 13 interviews were conducted. An overview of the interviews is presented in Table 4. The participants were a mix of both males (69.3%) and females (30.7%). All of the interviews were conducted between the 7th of February and the 3rd of March 2023. The interviews were carried out in Norwegian, as that is the first language of both the interviewer and all of the informants. For each interview, audio recordings were made. This was done in order to include as many details as possible and for the interviewer to focus on the conversation instead of transcribing during the interview. It is not recommended to rely on memory alone, as our memory is unreliable and prone to bias and error (Oates et al., 2022).

Case	**Role**	**Informant**
Case 1	Onboarder	P1
	Onboarding Leader	P13
Case 2	Onboarder	P2, P3
Case 3	Onboarder	P4, P9
Case 4	Onboarder	P5
Case 5	Onboarder	P6, P8
	Existing Team Member	P11, P12
Case 6	Onboarder	P7, P10

Table 4: Informants and the belonging role and case

3.4 Qualitative Data Analysis

The fourth step is *analysis of collected data*. It is an important step as it provides an in-depth understanding of the investigated cases (Runeson and Höst, 2019). It is also very suitable for case studies due to including everything except numeric analysis

(Oates et al., 2022). Figure 3 shows a BPMN model of the data analysis process based on Oates et al.'s (2022) suggestion on how to conduct qualitative data analysis.

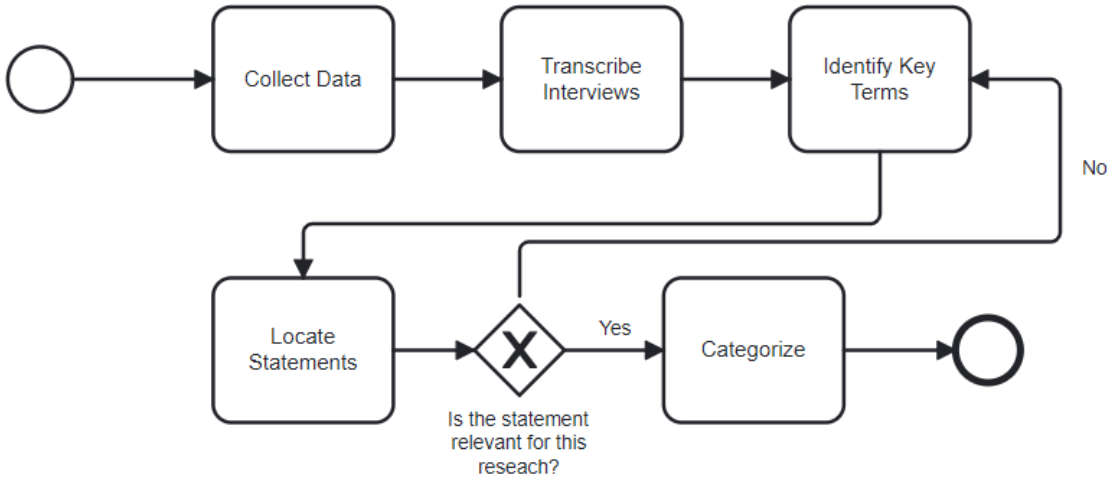


Figure 3: Process of the qualitative data analysis (Oates et al., 2022)

The first step *collect data* has already been presented in subsection 3.3. Next, the *interviews were transcribed* with help from the audio files and Microsoft Word transcription tool. First, the automated transcription was conducted by using the transcription tool. Then, all the audio files were listened to and compared to the transcribed text to make sure it was transcribed correctly. The automated transcription had an accuracy of approximately 50%, so a proper review was much needed. One reason for the low accuracy can be that the chosen language was Norwegian and not English. The most struggles with the transcription tools were on abbreviations and slang words.

The third step in Oates et al.'s (2022) suggestion to data analysis is to *identify key terms*. The step involved reading through all the gathered data to get an overall impression and discover the main findings. Then, in the next step, *locate statements*,

began. Statements irrelevant to this research were removed, at least for this particular study. The statements that provided a general description were the next to be located. This included the reason for working remotely, onboarding duration, onboarding strategy etc. The last statements to be gathered were other relevant data to this study. Further, in the last step, *categorize*, this data was divided into different categories depending on the context of the statement. Some of these categories were general information, agile principles and onboarding strategy. The categories were again divided into smaller parts before the next step in Oates et al.'s (2022) qualitative data analysis. The categories are presented in Figure 4. The figure does not include all details as each of the different onboarding strategies and agile practices

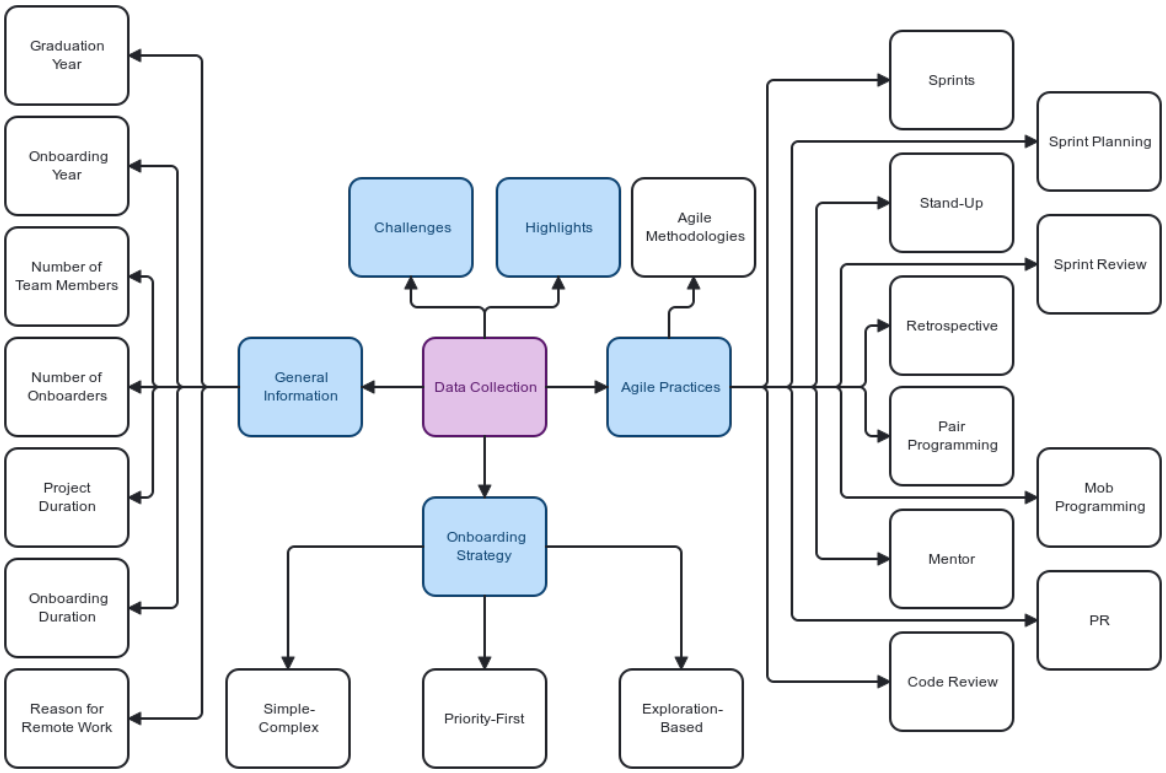


Figure 4: Categorised codes used in the data analysis

again was divided into the use, challenges and highlights for each of them. The qualitative data analysis tool, NVivo was used to code and analyse the collected data. This tool has created a great overview of the data, both regarding the categorisation, but also the be able to see the number of informants in each category.

After the collected data was analysed, the last step, *reporting*, started. The biggest challenge during this step was that the language had to be translated. As mentioned, the interviews were carried out in Norwegian and important parts of the interviews had to be translated into English, as this is the chosen language for this report. During all of the previous steps, the language had been Norwegian. The translation was not a straightforward process, as it is important that the context and meaning of the sayings do not change. The reporting and the results from the analysis are presented in section 4.

During the interview, the informants were asked if they wanted to have the opportunity to read the analysis of this study when done. However, only informants from three out of six cases wanted this opportunity. After the analysis was complete, it was sent to the interested informants. This was done in order for the informants to give feedback if they did not agree with the analysis result. The feedback the research received was that the informants agreed with the researcher's interpretation.

3.5 Method Evaluation & Limitations

3.5.1 Research Validity

According to Runeson and Höst (2019), the validity of a study indicates the trustworthiness of the results, to what extent the results are true and not biased by the

researcher's subjective point of view. This is something that needs to be considered throughout all the phases of a case study. There are several different ways to classify aspects of validity and threats to validity in the literature. However, this research has chosen to divide the section into four parts based on Runeson and Höst's (2019) four aspects of validity.

Construct Validity

The construct validity reflects on to what extent the operational measures that are studied really represent what the researcher has in mind and what is investigated according to the research questions (Runeson and Höst, 2019). Measures to ensure this validity include carrying out a pilot interview where an informant was asked to describe the interpretation of the questions and give feedback if anything was unclear. Furthermore, this feedback has been taken into account. Additionally, during the conducted interviews, important definitions were presented, so the interviewer and informant had the same interpretation. All the informants have also passed set criteria prior to the interview. These criteria are presented in subsection 3.3.

Internal Validity

The internal validity is of concern when causal relations are examined (Runeson and Höst, 2019). According to Yin (2009), this aspect is only relevant for explanatory or causal studies and not descriptive or exploratory studies.

External Validity

External validity is concerned with to what extent it is possible to generalize the findings and to what extent the findings are of interest to other people outside the investigated case (Runeson and Höst, 2019). In case studies, it is important not to make broad generalizations based on one action from a case that might have unique features not found in other situations (Oates et al., 2022). This research attempts

to give the reader sufficient information about the cases and the problem, so it is possible to make their own assessments about whether the finding can be applied in other settings. Additionally, this study attempts to reflect on how representative the selected problem situation is of other settings. However, a weakness of the external validity is having too few participants. If the research does not have enough participants it is impossible to show that a result is statistically significant (Oates et al., 2022). This is typically found in a case study (Oates et al., 2022). This study has six cases, which is more than enough to be a multi-case study but not enough to generalize uncritically.

Reliability

The reliability is concerned with to what extent the data and the analysis are dependent on the specific researchers (Runeson and Höst, 2019). The goal is that if another researcher had carried out the same study the results would have been the same. To achieve this, Yin (2009) suggests that the method and process should be documented. This report includes a detailed description of the entire research method, as well as the interview guide is added in Appendix B. Hopefully, due to this, another researcher will be able to use this and produce similar results.

3.5.2 Limitations

While the qualitative exploratory multi-case study approach employed in this master thesis offers valuable insights into the research topic, it is important to acknowledge certain limitations.

It has been shown that in interviews, informants can respond differently depending on how they perceive the interviewer, which means that the data generated can depend

on the perceived role and identity of the researcher (Oates et al., 2022). This is a possible limitation of this study. In addition to this, the findings are primarily based on subjective interpretations and perceptions of the onboarding experiences, which may introduce bias or subjectivity.

Interviewee-based research can be misleading since it focuses on what the informants say they do or think rather than what might really be the case (Oates et al., 2022). To mitigate the impact of individual interpretations and enhance the robustness of findings, it is crucial to utilize multiple data sources in a case study (Runeson and Höst, 2019). A conclusion is more credible than one that is solely based on one source of information if it can be drawn from several sources of data. While most cases in this study involved multiple informants, it is important to acknowledge that one case had only one informant, limiting the available perspectives and introducing potential biases.

The limitation of conducting all interviews digitally should be acknowledged. The absence of face-to-face interaction hindered the researcher's ability to establish a relationship with the informants prior to the interview and potentially compromised the opportunity to develop a deeper level of trust.

As a first-time case study researcher, there may have been overlooked or misinterpreted aspects of the applied method. However, the researcher will learn from any mistakes made in this study to improve future research endeavors.

Furthermore, the time and resource constraints imposed on the research restricted the depth and breadth of data collection, potentially leaving some important aspects unexplored.

4 Results

This section presents the results of the analysis from the case study described in section 3. All the results in this section are based on statements from the informants. As the research question implies, this study will mainly focus on three things regarding onboarding experiences; onboarding strategies, the use of agile practices in onboarding and onboarding success. The result is also divided into these three sections.

Before addressing the three main focus areas in this study, the onboarder's working arrangements needs to be presented. To what extent the teams have had a remote onboarding process varies slightly. Case 3 and 6 have worked remotely throughout the entire onboarding process. Case 1 and 4 have, throughout the project, worked physically together with the entire team one day a week. Case 2 and 5, had the first period as a team member at the office before everything started remotely. However, only in Case 5 the whole team actually joined these days physically. Table 5 presents the digital attendance of the onboarders.

Work Arrangements	**Case**
Fully remote	Case 3, Case 6
Remote after the first weeks	Case 2, Case 5
Hybrid	Case 1, Case 4

Table 5: The onboarders' working arrangements

4.1 Onboarding Strategy

As mentioned in subsection 2.2, there are three main ways to be assigned tasks when joining a team for the first time according to Ju et al. (2021). Based on the state-

ments regarding assignment of tasks during the interviews, the analysis has placed the onboarders within the different onboarding strategies. This section will present this placement and some experiences with the strategies.

Table 6 presents an overview of what onboarding strategies the onboarders have been onboarded within this study.

Onboarding Strategy	#
Simple-Complex	P6, P7, P10
Priority-First	P1, P5, P8
Exploration-Based	P2, P4, P9
None	P3

Table 6: Summary of the used onboarding strategies

Simple-Complex

Simple-Complex was the first onboarding strategy presented in section 2. The analysis has revealed that the onboarders from Case 6 and one of the onboarders from Case 5 were onboarded with a strategy similar to this (P6, P7, P10). They all mentioned that at the beginning of the onboarding, the tasks they were assigned were pretty simple and smaller than normal tasks. The tasks needed to be done but did not have a set deadline or a high risk.

In Case 5, P11 was the one that assigned the tasks to P6. According to P11, this strategy was used in order to learn the basic routines before the tasks' complexity increased. Just before P6 joined the team, simple tasks from the backlog were set aside so the onboarder could start with those.

The analysis has revealed that this strategy can be effective in building confidence among onboarders. It was observed that as the onboarding process progressed, the onboarders reported an increase in self-belief and a reduced sense of burden on the team. In contrast, developers from other strategies mentioned experiencing difficulties in feeling confident during the onboarding period.

Another finding is that the onboarders in this strategy needed a limited amount of training and coursing. There was learning enough in just working on assigned tasks as the complexity level of the tasks increased simultaneously with the knowledge level.

Priority-First

Priority-First is the next onboarding strategy. The onboarders that were assigned tasks based on task priority were from Case 1, 4 and 5 (P1, P5, P8). Straight away, these onboarders started on a critical task that any other developer on the team could do. Naturally, they got more help in the beginning (P1, P8).

The onboarding leader in Case 1 said that in their organization, they always want to use this strategy, as they want the onboarder to start to produce value for the team from day one. P1 agreed that this was how this onboarding strategy was perceived. Using this strategy was also a way for the team to show that they have faith in the onboarder, which P1 found *"pretty cool"*.

> *"Since my team used Scrum, I was thrown right into sprint planning on the first day. After this, I just followed the sprint and picked tasks from the backlog to do. It was very clear that they wanted to get me into work as quickly as possible."* (P1)

In Case 4, P5 mentioned that this method was chosen because of the team's constant and small releases. There was rarely any time for experienced developers to treat onboarders differently and find specific tasks for them.

The developers who onboarded with this strategy found it hard to ask questions to the experienced developers. The teams were under pressure, and therefore, it felt like they were disturbing others when they needed to ask for help. P1 mentioned that asking for help was like admitting to the team that you were not good enough to manage on your own. They often spend too much time on tasks and making up complex solutions to problems.

In Case 5, both Simple-Complex and Priority-First were used. In P6's case, Simple-Complex was chosen due to being able to learn the basic routines before the tasks' complexity increased. This was the team's normal approach when onboarding newly graduated developers. However, when P8 onboarded, a deadline was approaching and the team's normal onboarding routines had to be set aside. In addition to this, the team was using Scrum, so it was not a difficult process to change onboarding strategy. P11 participated in making a decision to change strategy and looking back, it was the right choice given the circumstances.

Exploration-Based

Exploration-Based is the last mentioned onboarding strategy. Onboarders from Case 2 and 3 onboarded with this strategy (P2, P4, P9). The onboarders experienced being assigned tasks that were not in the team's backlog. According to the onboarders in Case 3, this was done in order to explore on their own at their own pace before starting real and more critical tasks. P2 mentioned that the tasks were not very detailed and

a lot of research was needed to complete them.

P2's overall impression was that little thought was put into this strategy and that it was a sign of laziness and unpreparedness. This is also the impression after analysing the results. All other onboarders that used different onboarding strategies participated in presentations or courses at the beginning of the onboarding to get to know the team and the organisation and learn necessary work practices and technologies. The presentations and courses were arranged by both the organisation, but also by the other team members. In Case 2 and 3, the onboarders were tasked independently to learn the technologies and methodologies by reading and doing it.

> *"When it came to learning how the technologies worked and how the code was structured, it was up to us. We had to try and fail and were responsible for asking for help if needed."* (P9)

Both P2 and P3 are a part of Case 2. However, the analysis has revealed only one of them in this onboarding strategy. This is due to P3 feeling that no onboarding strategy at all was used because there were no tasks being assigned. P2 onboarded some time after P3. Within this time, the process has improved but still had a long way to go.

> *"They seemed to be hoping that I would contact them on my own to get something to do and that I knew what to do rather than that they had tasks to give to me. They somehow didn't know what to use me for."* (P3)

4.2 Agile Practices & Activities

As the research question in subsection 1.2 implies, only agile teams are researched in this report. However, what agile practices and activities the different cases use and how these have affected the onboarding process are different from case to case. This section will first talk about what agile practices the cases have been using before the different activities are presented.

Agile Practices

All the informants reported that either practices from Scrum, XP or a combination of those were used during the project. Case 1, 3, 5 and 6 were scrum teams of various degrees. Case 6 only used a limited version of Scrum, while the team in Case 1 even had a scrum master as a part of the team as the only case in this study. P5 identified the team in Case 4 as an XP team. This was due to a large number of tests, both unit and acceptance, the amount of pair programming and small and constant releases. The team in Case 2 is an abnormal case when it comes to working in a so-called agile team. The onboarders were informed in advance that the team would be using Scrum. However, when the onboarders joined the team, they noticed that almost no agile-related activities were used.

The onboarding leader (P13) mentioned that learning the onboarder the principles of the development methodology was a priority because when knowing the principles, it is easier for the onboarder to know what is needed to be done. The existing team members in Case 5 agreed with the onboarding leader. The onboarder in Case 1 mentioned the use of Scrum as one of the highlights of the onboarding process.

"Since I was already familiar with the principles of scrum from my studies, I really liked that I could just jump right into it. I think it is a lucid method, so it is very easy to know what is needed to be done, even when I am new. This made the days a little easier at the beginning." (P1)

Agile Activities

Table 7 presents which agile activity the different cases have been using during the onboarding process. As the table shows, the amount of activity varies from case to case. The colours in the table indicate to what extent the activities have been used (■ = To a large extent, ■ = To some extent, ■ = None).

Activity	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Sprints	■	■	■	■	■	■
Sprint Planning	■	■	■	■	■	■
Stand-Up	■	■	■	■	■	■
Retrospective	■	■	■	■	■	■
Sprint Review	■	■	■	■	■	■
Backlog	■	■	■	■	■	■
Pair Programming	■	■	■	■	■	■
Mob Programming	■	■	■	■	■	■
Mentor	■	■	■	■	■	■
Pull Request	■	■	■	■	■	■
Code Review	■	■	■	■	■	■

Table 7: Agile activities arranged in the different cases.

Sprint planning is an agile activity that the teams from all cases have performed at the beginning of each sprint to some extent. However, the onboarders from both Case 2 and 3 did not participate in it at the beginning of the onboarding. After some time on the team, Case 2 started having sprint planning. However, this did not actually help with the overall structure of the team's work. More changes than this were needed to fix the team's problems.

When it comes to having stand-up meetings, P1 said that although they worked from separate locations four days a week, they still had stand-up meetings simultaneously every day. Case 3, 5 and 6 have had the same arrangements; only it was fully remote. The members of the team in Case 4 did have stand-up meetings, but not every day.

> *"We had a slightly longer stand-up than what is perhaps normal so that we could catch up with each other about how things were actually going."* (P7)

Case 1 and 4 were able to attend the sprint reviews and retrospectives physically from the office during the onboarding process. The other onboarders also participated in these activities to some extent, but from a digital platform. In Case 5, the onboarders were asked to be responsible and take the lead. This was done on a request from the experienced developers as they wanted the onboarders to become more comfortable within the team and for them to learn the routines fast.

Naturally, the onboarders in the Priority-First onboarding strategy used a backlog and completed critical tasks from it. Several of the other onboarders also worked on tasks from a backlog, even though the onboarding strategies do not necessarily indicate it. In the Simple-Complex onboarding strategy, all onboarders used a backlog. However,

they did not work on the most critical tasks.

Pair programming has been mentioned as a good practice by informants in Case 3 and 5. Of the onboarders that have pair programmed, all of them have programmed it with the more experienced developers on the team. The existing members in Case 5 were a part of the pair programming as experienced developers. They mentioned that this activity was done in order to share knowledge and hopefully reduce time spent on tasks in the future. In addition to this, P4 also occasionally pair programmed with other onboarders. Even though they both were new, they knew different things and could learn from each other and complement each other well. Mob programming is another activity that Case 3 arranged.

> *"In the beginning, there was a lot of pair programming. This was a very good way of transmitting knowledge. Normally, someone with a lot of experience programmed together with me. It was very educational but also a bit stressful."* (P8)

Case 5 is the only team that used some kind of mentor during the onboarding. P11 and P12 were mentors for the onboarders in this case. This was not a set arrangement. It was the mentors' own initiative to take this role and to give the onboarders close follow-up.

> *"I am very satisfied with the onboarding based on the situation [Covid-19 pandemic]. I had good follow-up from one person, and that was all I felt I needed."* (P8)

The analysis has revealed that several of the other onboarders would have wanted

to have a mentor or someone experienced available for them. In Case 2, they were promised a mentor but never got one. The onboarding leader (P13) mentioned that having that one person to interact with is especially important when it is a young developer onboarding. After gaining more experience, the need for a mentor reduces. The onboarders in Case 3 said that the follow-up the first couple of days was very good, but after some time, they felt very much alone.

> *"It was possible to ask the team for help, but I think the threshold was too high. It would be great to have a mentor who could have been a little more present to help me or who called and checked if things were going well from time to time."* (P4)

Creating pull requests (PR) and using code reviews is an action that most of the informants have been a part of. P13, the onboarding leader, mentioned that they normally have restrictions on who can approve PR during onboarding. In Case 1, 4 and 5, the more experienced developers had to approve the PRs during the onboarding. However, the onboarders in Case 5 were able to contribute to the code reviews and give feedback to the other team members. After some time, they were also able to approve others' PR. Code reviews have also been a great way of learning from experienced developers without having to interrupt them. All the informants that participated in PR's or code reviews were positive to the use of it.

> *"I was already allowed to take part in a major PR in the first week. Even though I felt very insecure and still was a newcomer to the team, I was allowed to have a PR of 1000 plus lines. It's cool."* (P5)

Lastly, the analysis has found that all of the onboarders in this study have worked

independently to some extent. Onboarders from Case 1, 2, 3 and 6 noticed that, especially after the first couple of days, a lot of the work was completed alone. P2 from Case 2 said that the longing for interaction with other people got greater throughout the onboarding process. P4 agreed with this and felt really lonely during the first experience as a professional software developer.

Communication

As mentioned in section 2, communication and collaboration are important parts of agile software development. Based on the analysis, it has been discovered that the different cases have chosen quite similar forms of communication during the onboarding processes. Most cases mentioned that Slack was used a lot for both academic and social purposes. In Case 1, 4 and 5, Slack was the main source where they contacted the other team members when there was a need for help with anything. Slack was used all the time, every day. Most of the Slack channels were for work-related purposes, but some were mainly used to joke around and plan social activities.

> *"The positive thing about remote onboarding is that the threshold for sending a message on Slack has become very low. It has made it much easier to ask for help, even when working physically in the office. It is often easier to send a message asking "stupid" questions than to ask physically."* (P1)

While Slack is mostly for written communication, other tools were used when video and audio were needed. However, it can be hard to express uncertainty when using digital communication. Body language is very important in such a vulnerable situation and does not come across in the same way as in a physical environment, according

to P5. Informants in Case 1 and 4 said it has helped a lot to work physically one day a week to improve team communication. All the cases in this study mentioned that working more physically together would have helped to make things easier when it comes to communication and collaboration.

> *"It is tricky when you can not point to the screen and ask, "What is happening here?" in the same way when working from different locations."* (P5)

P7 said it was surprisingly easy to talk to people to ask for help when working remotely. Asking questions did not affect the concentration of the other team members the same way as before because it is possible to turn off notifications.

Social Arrangements

Social arrangements include activities that aim to help with an agile team's social aspect. According to the informants in this study, the most common challenge in remote agile teams is getting to know the other members of the team and collaborating in a good way. P4 and P13 believe that a better social environment within the team can help the team to be more successful and that a more social onboarding process could help to achieve this.

In all cases, non of the social arrangements have actually been arranged due to the onboarding processes. However, except for Case 2 and 6, all onboarders have participated in some kind of either physical or digital social arrangements during the onboarding. Both Case 2 and 6, said that when the lockdown was over, they started to arrange more social events, but this was after the onboarding process was over.

The most commonly mentioned social activity was different types of after-work (quiz, *"lønningspils"*, having dinner, playing games etc.). During working hours, onboarders from Case 1, 3, 4 and 5 have mentioned having coffee talks, lunches and check-ups. However, in Case 1, 4 and 5, there has been poor attendance among the more experienced and older developers in the activities. This has contributed to the fact that getting to know the team took longer. The members of Case 3 and have still, to this day, not met any team members in person. Onboarders from Case 1, 2, 3 and 6 said that they would have wanted to have more social arrangements.

Since it can be difficult to get to know others in the company when working remotely, it is important to get along with the team (P13). P5 could relate to this and was lucky to get along well with the team. The onboarder has also taken the initiative to socialize with others in the team several times (P5). This made for a very good atmosphere in the team. Sometimes one person can make the whole difference in a team. In Case 5, there was a struggle for the onboarders to get to know the whole team. However, the onboarders were lucky to have two people who followed up, and they became good friends. It was this people that made P6 and P8 think the onboarding was an overall good experience.

4.3 Onboarding Success

This section aims to present the success of the onboarding processes based on the analysis of the interviews. There has been a varying degree of success among the onboarders. However, all the informants in this study have experienced challenges or have something they would have wanted differently in their onboarding process.

At the end of the interviews, the onboarders were asked a question about whether

they were happy with the onboarding. Naturally, for most of them, it was not as easy as a yes or no answer. However, in the end, P1, P5, P6, P7, P8 and P9 claimed to be happy with the overall process. In Table 8 and Table 9, an attempt has been made to present the success and failure factors of the different onboarding processes based on the analysis. The success or failure factors are quotes from the onboarders that the analysis believes sum up the main reason for the onboardings to be a success or not.

#	Case	Success Factor
P1	1	*"I think it was the use of Scrum and the fact that there are rather strict guidelines on how to complete the work that made this a good onboarding for me."*
P5	4	*"The highlight of the onboarding has probably been the people on the team. And especially those from the same company as me. We have become a very nice group."*
P6	5	*"I was lucky to have a mentor. I believe he made the whole onboarding for me."*
P7	6	*"I felt that since we had social breaks and I got to talk to a few people during the working day, it worked well for me. I feel like I have got a good structure on the work day, and I really like my team, so."*
P8	5	*"I am very satisfied with the situation. I had good follow-up from two people and that was really what I felt I needed."*
P9	3	*"It worked for me since I like to work independently and I do not mind not seeing anyone throughout the work day."*

Table 8: Success factors of the successful onboardings

#	Case	Failure Factor
P2	2	*"What I missed most was people who knew what I was doing and to whom I could ask questions. It would have helped a lot."*
P3	2	*"Everything should have been done differently."*
P4	3	*"If there had been more physical attendance, it would have been easier to get to know the team and, my threshold for, e.g. asking for help might have been lower. This could have made me even more productive and learn faster."*
P10	6	*"I feel like having the opportunity to meet my team and having better arrangements for communication flow is something I would have wanted differently."*

Table 9: Failure factors of the failed onboardings

The interviewed onboarding leader and the existing team members were all happy with the onboarding processes regarding P1, P6 and P8. They mentioned that the onboarders learned all that was needed within the expected time.

That the onboarders don't feel like they could complain because others had it worse than them was a common statement in the interviews, especially among the Covid-19 affected teams.

> *"I had some luck. I onboarded during a period when people who had already had horrible onboarding processes were part of the team. So much had improved."* (P5)

In Case 5, the existing team members were a part of the team when the Covid-19

pandemic started, and the team had to transfer to become a remote team. This was a tough process. Trying to transfer the onboarding process to a remote environment on short notice without any previous experience led to many challenges. Luckily, most of the challenges were overcome, and the onboarding ended up being a success.

As mentioned, the most commonly reported wanted change to help the onboarding be more successful was to have more follow-up, more physical attendance and more social arrangements. The reason for wanting these changes has been a bit different depending on the onboarder.

The analysis of the results revealed that when the onboarders were asked to describe the most important events in the onboarding process briefly, the onboarders rarely mentioned onboarding strategies or agile-related activities or practices. They automatically thought about the first couple of days working on the team. They explained how they were introduced to the team, the different technologies, the project and the organisation they were working for.

4.3.1 Onboarding Duration

The onboarding leader mentioned that based on previous experiences with onboarding, it takes approximately three months for an onboarder to start to provide any value to the project and the team. After this time, it has been normal for the onboarders to give more to the team than they take. The other cases in this study can help to strengthen the onboarding leader's statement about the onboarding duration. With the exception of P3, P5 and P7, everyone agreed that the onboarding was complete after two to three months. In Case 5, the existing team member confirmed the onboarding duration for the onboarders. The onboarding duration is presented in

Table 10.

P1 already knew that this was a contemporary team when the onboarding started.

> *"... Because of this, I never felt like a proper member of the team. However, after working with the team for two and a half months, I was given the opportunity to be responsible for onboarding other newcomers, which can mean that others looked at me as fully onboarded."* (P1)

P3, from Case 2, did not feel like the onboarding was complete after three months. The onboarder claimed never to become a proper team member and never quite understood what the project actually was. Therefore, P3 switched teams before any proper value was created for the team. P5 and P7 felt like the onboarding process was shorter than three months. P5 said getting to know the code and the other team members was very easy. Therefore, P5 claims to be onboarded completely after only three weeks. P7 had worked on the same project during a summer internship and, therefore, already knew some team members and were familiar with parts of the code.

Onboarding Duration	#
$<$ 3 Months	P5, P7
$\approx$ 3 Months	P1, P2, P4, P6, P8, P9, P10
$>$ 3 Months	P3

Table 10: Summary of the onboarding duration for each onboarder

However, it should be mentioned that P1, P2, P3, P4 and P10 never felt like proper team members during their first experience as consultants on a project, even though they might have completed the onboarding process.

5 Discussion

This section will discuss the results in section 4 against the research question presented in section 1.2. The research question includes looking into remote onboarding of newly graduated software developers to already existing agile teams. The goal is to map out experiences with the most common onboarding strategies, how agile principles and activities can contribute to an onboarding process and the success of the onboarding processes. This section is divided into these three parts starting with onboarding strategies, followed by agile practices and activities and the success of onboarding.

5.1 Onboarding Strategies

The analysis of the results shows that there was an even distribution of onboarders on the different onboarding strategies, which makes the basis for discussion the same for all strategies.

According to Ju et al., an Exploration-Based onboarding strategy should mainly be used on experienced developers or on newly created teams. In this study, neither was the case. The onboarders all had little to no experience, and the project duration at the time of the onboarding was between six months to over a year. This strategy also makes the onboarders work outside the team's core production. Therefore, it might be particularly important with efficient onboarding since the work does not contribute to the product being developed. It should also be mentioned that since onboarders work outside of the team's normal work routines, these routines have to be learned as well later on.

Based on the three onboarding strategies, the onboarders in the teams using this approach expressed the lowest level of satisfaction. This low satisfaction was reasoned by a large amount of individual work and having trouble getting to know the other team members. These challenges can not automatically be fixed by changing the onboarding strategy but rather by changing the use of agile and social activities. Nevertheless, based on this study, the Exploration-Based onboarding strategy should not have been used in the first place, and this study contributes to confirming this.

Additionally, an example from this strategy can show that similar onboarding processes can be experienced in different ways depending on the developer. Both onboarders in Case 3 were onboarded within the same team with the same strategy, completed the onboarding within the same duration and participated in the same activities. However, when it comes to how they thought the overall experience was, they disagreed. One of them was happy with the onboarding process, while the other never felt like a part of the team and felt very much alone. Moe et al. (2020) discovered the same thing. They wrote that the outcome of an onboarding process could be different even if an organization applies the same practices and strategies for all the onboarders. Another researcher, Buchan et al. (2019), found out that onboarding processes should be adapted to each of the given scenarios and onboarder to get the best possible experience. The researcher in this study agrees with this based on the analysis.

The Simple-Complex strategy was a greater success among the onboarders in this study. As the analysis implies, this onboarding strategy can help to build the confidence of the onboarders and help them gain the necessary knowledge without a lot of extra training and courses. In addition to this, it is also the onboarding strategy Rodeghero et al. (2021) and Buchan et al. (2019) recommended to use when

onboarding newly graduated developers.

Using this onboarding strategy does not mean that the use of agile practices and activities has to be affected, as mentioned by Gregory et al. (2022). All of the onboarders in this strategy used Scrum as a methodology and completed tasks from the backlog.

P10, who expressed some dissatisfaction with the onboarding process, highlighted that increased physical attendance and improved arrangements for asking questions would have contributed to a better onboarding experience. P10 also noted that not participating in any social activities with the team may have contributed to the dissatisfaction. It is important to note that these challenges are primarily related to working remotely and the overall implementation of agile practices and activities, rather than the specific onboarding strategy.

The analysis of this study can help to show that the Simple-Complex strategy can be a good choice when onboarding newly graduated software developers. It can help to build the necessary confidence and quickly learn the team's routines.

Gregory et al. (2022) mentioned that the use of the Priority-First onboarding strategy could be a great choice in agile software development teams, especially when under pressure. The results of this study agree with this. The onboarders were introduced to agile activities and started to produce value for the team quickly. However, it is important to provide good follow-up to the onboarders when using this strategy. The onboarders mentioned that since the team was under pressure, it was difficult to ask questions. One of the onboarders had a mentor, and this onboarder said that the onboarding would not have been the same and probably not have been successful without this follow-up.

Based on the results, the Priority-First onboarding strategy can be successfully used on newly graduated developers even when the team is not under pressure if the necessary follow-up is provided.

One of the onboarders was onboarded without a given strategy. The same developer is also the only informant who did not complete the onboarding within three months and the one that would have wanted everything to go differently. Although this is only one onboarders' experience and opinion, it can help show that it is necessary for newly graduated software developers to be onboarded with a plan and a strategy for it to be a successful process. Rodeghero et al. (2021) and Moe et al. (2020) both recommended that creating a customized plan was a good measure.

5.2 Agile Practices & Activities

There are a lot of different practices and activities to look into, and figuring out how each of them has affected the different onboarding experiences would have required a larger amount of data. However, there are some interesting findings to discuss.

First of all, Case 1, 4, and 5 were the ones that used agile practices to the greatest extent. All onboarding processes in these cases have been a success, according to the informants. More valid data is needed to conclude anything. However, it can indicate the beginning of a trend that shows that the use of agile practices and activities will help to make the onboarding of newly graduated developers successful.

There are several different challenges regarding onboarding processes with similar solutions in this study. These challenges are, for instance, feeling lonely or not feeling like a part of the team, having problems asking questions or feeling like you are

disturbing your team members or not knowing your team members. By facilitating more social activities, having better follow-up, assigning a mentor or suggesting pair or mob programming, these challenges can be reduced. Based on this study, the key is to participate in and facilitate academic and social activities that help to improve communication and collaboration within a team in order to improve the overall onboarding process.

When it comes to having a mentor, several studies recommend using it, and they also point out how extremely valuable it can be for onboarding success (Bauer and Erdogan, 2011; Buchan et al., 2019; Gregory et al., 2022). However, the results show that only two onboarders got a mentor when onboarding. They were both really satisfied with this, and both of them claimed this to be a big reason for the success of the entire onboarding process. In addition to this, having a mentor was one of the most mentioned wanted changes. These findings can help to contribute, along with the other studies, to show that having a mentor is an important factor in achieving a good onboarding experience for the onboarder. It should also be mentioned that a mentor should get the necessary training and be presented with have is expected from them (Buchan et al., 2019).

Buchan et al. (2019) also carried out a study on remote onboarding to agile teams. From this study, five onboarding techniques were recommended to consider for a successful onboarding process. These onboarding techniques are socialisation opportunities, access to high-quality knowledge artefacts, access to formal training, proactive feedback, provide psychological safety. The analysis supports several of these recommendations. Having a mentor, participating in social arrangements and building an environment where questions can be asked were mentioned as some of the most important measures. This corresponds to Buchan et al.'s (2019) recommendations

regarding socialisation opportunities, proactive feedback and access to formal training.

It was also noticed that older and more experienced developers tend to participate less in social events. This can have consequential effects on other aspects of the onboarding process. Since onboarders rely on learning from experienced developers, it is crucial for them to make themselves available and accessible. In a physical office setting, opportunities for interaction and getting to know each other naturally arise. However, in a remote work environment, alternative social activities are necessary to foster these connections and facilitate knowledge transfer.

Every onboarder worked independently to some extent in this study. However, it is interesting to look at the amount of independent work and the reason for the onboarding being a success or not. The main reason for not being happy with the onboarding was the lack of relationship and safety within the team, and one of the reasons for being pleased with the onboarding was specific team members or the team in general. The onboarders with the most amount of independent work are the same as the ones wanting a better relationship within the team. This can contribute to showing that during the first time on a new team, the amount of independent work in agile teams should be limited.

Even though the onboarders in Case 2 were told that the team used agile practices and activities, this was not the case. The team had no plan or structure and used almost no agile activities in their work. In addition, these onboarders were not pleased with the onboarding process. They missed having a plan or at least having someone telling them what to do and when to do it. Using agile practices would have contributed to this. Based on this study, it is suggested that by integrating some routines and agile practices into this team, the onboarding processes and the project, in general, would

have been a greater success.

Taft's (2007) findings regarding the onboarding of newly graduated software developers included that onboarders commonly lack communication and teamwork skills, are unprepared for complex development processes, legacy code, and deadlines, and work with limited resources. Based on the analysis, the fact that the onboarders lack communication and teamwork skills might be true. Many of the discovered challenges concern this. When it comes to the other parts of the findings, the results in this report can neither confirm nor decline them.

5.3 Onboarding Success

Discussing the success of onboarding can be difficult. It can vary greatly from developer to developer, and there are many factors that must be taken into account. Every onboarder in this study has experienced challenges or has something they would have wanted differently in their onboarding process. In this study, when the onboarders were asked the simple question if they were happy with the overall onboarding process, the question did not mention on what basis it should be answered. Therefore, this question does not say anything about if the onboarded have gained all the necessary knowledge or completed the onboarding within the time expectation of the team.

However, Bauer and Erdogan (2011) tried to define onboarding success from an organisation's perspective by using four building blocks called Four C's. The Four C's are compliance, clarification, culture and connection. Basically, this means that the onboarding is not successful until all the building blocks have been achieved. Based on the analysis of the results, several of the informants never achieved all of these blocks. P1, P2, P4 and P10 probably never established the last building block *con-*

nection. They never felt like proper team members and had problems connecting with the team. P3 only managed to fulfil the first building block, compliance, if even so. However, P1, P2 and P4 still mention an end time of the onboarding, which indicates that they felt the onboarding was complete after some time, even though they did not achieve all blocks and were not necessarily happy with the process.

It is also worth discussing the reasons behind the different cases working remotely, as these reasons can significantly influence the onboarding experience and satisfaction. When onboarding remotely due to long commutes, it is usually a matter of choice. This means that if team members wanted to meet each other at the office, arrangements could be made. The informants who were onboarded to distributed teams worked digitally from different locations, often far apart. In these situations, it is typical for individuals to be aware that they are joining a remote team and understand that they will not have the opportunity to meet and work in person. This was the case in our study. Lastly, some informants were onboarded into remote teams due to the Covid-19 pandemic. This was likely not a matter of choice. During that time, there were many other concerns that took precedence over the team and the project being developed. The pandemic also introduced unpredictability, making it challenging to determine if the informant would ever meet the rest of the team or for how long the team would continue working remotely.

All of these different reasons and prerequisites will affect the attitude of the developers. If a newly graduated developer has a set of expectations for the onboarding process and the plan suddenly changes, the developer can be disappointed, and their expectations can be hard to fulfil. This can lead to the onboarder being unhappy with the onboarding process even though, with other expectations, the onboarder would have been pleased with the same arrangements.

The number of remote teams increased significantly when the Covid-19 pandemic started, and many teams had to move their work home instantly. This might have led to some of the developers onboarded during this period being the first-ever to onboard that team remotely. This implies that the team did not have a lot of previous experience with remote onboarding and might not even know what working remotely entails. Several of the informants who were onboarded during this period may, in many ways, have been pilot subjects for this type of onboarding for many companies and teams. This was the case in both Case 2 and 5. They said that they did not feel they could complain because someone else had it worse. This may be reflected in the results and may also indicate that onboarding processes automatically improve in the future as more and more experience is gained among software developers and onboarding leaders.

5.4 Limitations

The scope limitations and threats of research validity are already presented in section 1.3 and 3.5. This section will address factors that may still have had an impact on the result and which constitute the limitations of the research.

First of all, this is a multi-case study where the results and data are collected from a limited number of informants and cases. In a case study, multiple informants in a case are preferred (Oates et al., 2022). One of the cases is this study only has one informant, and several of the others have two informants. These numbers should probably have been higher, but due to time limitations, it was not possible. Even though there is a limited number of cases, six cases is a lot for a study of this size. This makes it difficult to go in-depth as much as one probably should on each of the

cases. Based on this, another qualitative research method might have been a better choice for this study. However, case studies provide an in-depth and rich description and understanding, which can provide useful data for further research.

The collected data is based on individuals' previous experiences and thoughts. Given that the study is gathering data pertaining to past events, there is a possibility of inaccuracies in the data's precision. This is due to the potential for informants to recollect information incorrectly or retain false memories. Consequently, the ability to validate the inferences being drawn may be constrained. Another important factor in this study is who the informants are and what their prerequisites are. However, this has not been taken into account to a large extent. This is a limitation. Additionally, since case studies allow the researcher to show complexities of life and to explore alternative meanings and explanations (Runeson and Höst, 2019), it is crucial to acknowledge that the researcher's personal perspective may impact the results.

In section 2, a theoretical foundation for this study is presented. The theory is not a summary of all available data on this topic, but a selection that the researcher found useful to provide a general understanding of the topic and to increase the basis for the discussion. Therefore, there might be additional relevant and interesting research out there that could have been used in this research as well.

It should also be mentioned that the researcher has little to no previous experience with case studies or carrying out interviews with the aim of collecting qualitative data. It is known that case studies require experience to be carried out (Oates et al., 2022), and limited experience can impact the data collection method implementation negatively. This is a possible limitation of the results. In addition to little experience with conducting research of this type, the researcher also only had a limited overview of the area of research in advance. As a consequence, the extent and profundity of

the discussion, as well as the comprehensiveness of the incorporated literature and theory, might have been reduced to some degree in comparison to the work produced by experienced scholars or researchers.

6 Conclusion

As perhaps expected, there are very different experiences when it comes to the processes of onboarding newly graduated software developers to their first-ever professional agile development team. Some informants have mentioned that everything should have been done differently, while others have all in all been satisfied with the onboarding process. Yet, every informant in this study has experienced challenges or has something they would have wanted to happen differently in their onboarding process. This can imply that this was a needed field to study and that more research should be carried out in this area in order to improve the process of onboarding software developers to already existing agile teams.

Even though there are some limitations to this study, several interesting findings have been discovered. Below are the main findings related to the utilization of strategies and agile practices and activities in onboarding, as well as their impact on the success presented.

- Both the Simple-Complex and Priority-First onboarding strategies are good options when onboarding newly graduated developers to remote agile teams. Simple-Complex is a good choice to build confidence and learn the team's routines quickly. Priority-First is the best onboarding strategy to use if the team is under a lot of pressure, given that necessary follow-up is provided. The Exploration-Based strategy is not the best choice for existing team onboarding newly graduated developers.
- Creating a plan and having set goals for the assignment of tasks for the onboarder is the most important thing when it comes to onboarding strategies.

- Using agile practices and activities to a larger extent can support the onboarding process in a good way.
- Having experienced developers to interact with through mentorship or collaborative activities, like pair programming or mob programming, can be an important success factor throughout the onboarding.
- Participating in social arrangements can help the team academically by creating a safer environment for the onboarders and improving communication and collaboration within the team.
- Similar onboarding experience can vary from developer to developer, and therefore should each onboarding process be customized to the current onboarder.

To conclude, the findings of this study indicate that there are several good measures and suggestions for achieving successful onboarding through the implementation of onboarding strategies and the use of agile practices and activities. Nevertheless, it is important to remember the value of having a well-defined plan while at the same time acknowledging that the onboarding experience can differ between onboarders. Therefore, it is essential to adapt the onboarding approach to each situation.

6.1 Further Work

According to Runeson and Höst (2019), an important part of exploratory case studies is to address new directions for further research. This section will aim to do so.

The findings regarding onboarding strategies align with existing research, except for the Exploration-Based strategy. In this study, the selected cases did not have well-

designed arrangements for this strategy. As a result, further research could explore this aspect in more detail to gain valuable insights.

This study's findings indicate a possible trend regarding the relationship between the use of agile practices in the onboarding process and the process's success. However, the volume of data is constrained. Therefore, it would be a really interesting approach to collect more information on this subject to determine whether increased use of agile practices and activities can actually help with onboarding success. Additionally, there is also a need for more data to examine the effects on onboarding of the various agile activities on their own.

The main purpose of onboarding is to become a proper member of the team and start to produce value for the team. The scope of this study is from the onboarder's first day of a new project until the onboarding is completed after a couple of months. There is no information about what happens after the onboarding. An interesting suggestion for future work is to follow the onboarder after the onboarding is complete and look into if proper value is actually provided to the team. Whether or not the onboarding was actually successful can depend greatly on this.

References

Bacchelli, A., & Bird, C. (2013). Expectations, outcomes, and challenges of modern code review. *2013 35th International Conference on Software Engineering (ICSE)*, 712–721. https://doi.org/10.1109/ICSE.2013.6606617

Barrero, J. M., Bloom, N., Buckman, S., & Davis, S. J. (2023). Swaa february 2023 updates - wfhresearch.com. https://wfhresearch.com/wp-content/uploads/2023/02/WFHResearch_updates_February2023.pdf

Bauer, T. N., & Erdogan, B. (2011). Organizational socialization: The effective onboarding of new employees.

Beck, K. (1999). Embracing change with extreme programming. *Computer*, *32*(10), 70–77.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Agile manifesto*. Retrieved January 25, 2023, from https://agilemanifesto.org/

Begel, A., & Simon, B. (2008). Novice software developers, all over again. *Proceedings of the fourth international workshop on computing education research*, 3–14.

Bell, J. T. (2017). Extreme programming. https://pdfs.semanticscholar.org/0018/95a221312d80f69bc3c525ff458d727fd0c5.pdf

Britto, R., Cruzes, D. S., Smite, D., & Sablis, A. (2018). Onboarding software developers and teams in three globally distributed legacy projects: A multi-case study. *Journal of Software: Evolution and Process*, *30*(4), e1921.

Buchan, J., MacDonell, S. G., & Yang, J. (2019). Effective team onboarding in agile software development: Techniques and goals. *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 1–11. https://doi.org/10.1109/ESEM.2019.8870189

DeMarco, T., & Lister, T. (2013). *Peopleware: Productive projects and teams.* Addison-Wesley.

Deshpande, A., Sharp, H., Barroca, L., & Gregory, P. (2016). Remote working and collaboration in agile teams.

Eurostat. (n.d.). https://ec.europa.eu/eurostat/web/structural-business-statistics/information-on-data/small-and-medium-sized-enterprises

Gregory, P., Strode, D. E., Sharp, H., & Barroca, L. (2022). An onboarding model for integrating newcomers into agile project teams. *Information and Software Technology*, *143*, 106792.

Ju, A., Sajnani, H., Kelly, S., & Herzig, K. (2021). A case study of onboarding in software teams: Tasks and strategies. *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 613–623. https://doi.org/10.1109/ICSE43902.2021.00063

Kraut, R. E., & Streeter, L. A. (1995). Coordination in software development. *Communications of the ACM*, *38*(3), 69–82.

Marek, K., Wińska, E., & Dabrowski, W. (2021). The state of agile software development teams during the covid-19 pandemic. *Lean and Agile Software Development: 5th International Conference, LASD 2021, Virtual Event, January 23, 2021, Proceedings 5*, 24–39.

Meier, A., Kropp, M., & Perellano, G. (2016). Experience report of teaching agile collaboration and values: Agile software development in large student teams. *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, 76–80. https://doi.org/10.1109/CSEET.2016.30

Micaela, B. (2020). Telework in the eu before and after the covid-19: Where we were, where we head to headlines.

Moe, N. B., Stray, V., & Goplen, M. R. (2020). Studying onboarding in distributed software teams: A case study and guidelines. In *Proceedings of the evaluation and assessment in software engineering* (pp. 150–159).

NimbleWork. (2023). https://www.nimblework.com/agile/extreme-programming-xp/

Oates, B. J., Griffiths, M., & McLean, R. (2022). *Researching information systems and computing*. Sage.

Pham, R., Kiesling, S., Singer, L., & Schneider, K. (2017). Onboarding inexperienced developers: Struggles and perceptions regarding automated testing. *Software Quality Journal*, *25*(4), 1239–1268.

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*, 303–337.

Pmadmin. (2023). What is scrum?: The agile journey with pm-partners. https://www.pm-partners.com.au/the-agile-journey-a-scrum-overview/

Rodeghero, P., Zimmermann, T., Houck, B., & Ford, D. (2021). Please turn your cameras on: Remote onboarding of software developers during a pandemic. *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 41–50. https://doi.org/10.1109/ICSE-SEIP52600.2021.00013

Runeson, P., & Höst, M. (2019). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, *14*(131).

Snell, A. (2006). Researching onboarding best practice: Using research to connect onboarding processes with employee satisfaction. *Strategic HR Review*, *5*(6), 32–35.

Sommerville, I. (2011). Software engineering 9th.

Taft, D. K. (2007). Programming grads meet a skills gap in the real world. *Retrieved September*.

Tellis, W. (1997). Introduction to case study. *The qualitative report*, *3*(2), 1–14.

Viviani, G., & Murphy, G. C. (2019). Reflections on onboarding practices in mid-sized companies. *2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, 83–84.

Wu, L., Waber, B. N., Aral, S., Brynjolfsson, E., & Pentland, A. (2008). Mining face-to-face interaction networks using sociometric badges: Predicting productivity in an it configuration task. *Available at SSRN 1130251*.

Yin, R. K. (2009). *Case study research: Design and methods* (Vol. 5). sage.

A Privacy Agreement

Personvernavtale

Avtale mellom

Student ved NTNU: Juni Leirvik Larsen	Fødselsdag: xx.xx.xxxx
Systemutvikler:	Fødselsdag:

Formål
Dette forskningsprosjektet er en masteroppgave på studiet MIDT (Master i Datateknologi) ved NTNU i Trondheim. Formålet med denne datainnsamlingen er å få bedre innsikt i valgt tema, slik at en bedre konklusjon kan bli lagd.

Hvem er ansvarlig for forskningsprosjektet?
Forskningsprosjektet utføres av Juni Leirvik Larsen som er student ved IDI, NTNU Trondheim. Dette gjøres i samarbeid med veileder, Torgeir Dingsøyr, som er ansatt ved IDI, NTNU. De er begge ansvarlige for dette prosjektet.

Hvorfor er du inkludert i studien?
Du er inkludert i studiet på grunnlag av at du sitter på noen erfaringer som er interessante

Hva innebærer dette prosjektet for deg?
For deg innebærer dette prosjektet å gjennomføre et intervju på 30-45 min hvor lydopptak vil bli tatt. Det er mulig det kommer oppfølgingsspørsmål i ettertid om nødvendig om du aksepterer det. Spørsmålene vil omhandler dine personlige erfaringer, så ingen svar er feil.

Du kan protestere
Du kan når som helst protestere mot at du inkluderes i dette forskningsprosjektet, og du trenger ikke å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du velger å protestere.

Ditt personvern – Hvordan vi oppbevarer og bruker dine opplysninger
Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

Navnet og kontaktopplysningene dine vil erstattes med en kode som lagres på egen navneliste adskilt fra øvrige data. Alt materialet vil lagres kryptert på institusjonens server. Det vil kun være student og veileder som har tilgang på materialene.

Hva skjer med opplysningene ved forskningsprosjektets avslutning?
Opplysningene vil være anonymisert ved leverandre av prosjektet. Alle andre filer og opptak vil slettes ved prosjektslutt. Antatt prosjektslutt er 5. juni.

Hva gir oss rett til å behandle personopplysninger om deg?

Vi behandler opplysninger om deg fordi forskningsprosjektet er vurdert å være i allmennhetens interesse, men du har anledning til å protestere dersom du ikke ønsker å bli inkludert i prosjektet.

På oppdrag fra IDI, NTNU Trondheim har Sikt – Kunnskapssektorens tjenesteleverandørs personverntjenester vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Dine rettigheter

Så lenge du kan identifiseres i datamaterialet, har du rett til:

- å protestere
- innsyn i hvilke personopplysninger som er registrert om deg
- å få rettet personopplysninger om deg,
- å få slettet personopplysninger om deg, og
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger.

Hvis du har spørsmål til studien, eller ønsker å vite mer eller å benytte deg av dine rettigheter, ta kontakt med:

- NTNU ved Torgeir Dingsøyr (torgeir.dingsoyr@ntnu.no).
- Vårt personvernombud: Thomas Helgesen (thomas.helgesen@ntnu.no)

Hvis du har spørsmål knyttet til vurderingen av prosjektet som er gjort av Sikts personverntjenester, kan du ta kontakt med:

- Personverntjenester på epost (personverntjenester@sikt.no) eller på telefon: 73 98 40 40.

Med vennlig hilsen

Torgeir Dingsøyr	Juni Leirvik Larsen
(Forsker/veileder)	(Student)

……	……
Student ved NTNU	Tid & Sted
……	……
Utvikler/Intervjuobjekt	Tid & Sted

B Interview Guide

Intervju Guide

Intro

Kort om meg

Studerer datateknologi retning programvaresystemer ved NTNU Trondheim og skriver for øyeblikket masteroppgave.

Oppgaven
Oppgaven jeg skal skrive handler om remote onboarding av nyutdannede i sitt første agile software development team. Målet er å finne ut hvordan remote onboarding gjennomføres av team for å videre finne ut av hvordan det burde gjennomføres for å få best mulig resultat slik at prosjektet blir så vellykket som mulig. I denne oppgaven så har jeg valgt å definere onboarding perioden hvor man går fra å være en outsider til å bli en insider i teamet.

Databruk
Dataen som blir samlet inn i dette intervjuet, vil bli brukt i et case-studium, hvor dataen blir analysert og sammenlikner med allerede eksisterende studier.

Praktisk

Tid
30-45 min

Lydopptak
Er det ok at det tas lydopptak? Lydopptaket vil kun bli brukt for å hjelpe egen hukommelse og notater. Personopplysninger vil anonymiseres og lydopptaket vil slettes ved prosjektslutt (ca. 5. juni). Om det er ønskelig å få tilgang på lydopptaket, er dette mulig.

Vennligst signer dette dokumentet før vi fortsetter. **Signer samtykkedokument**

Svar
Jeg ønsker at alle svar skal være ærlig. Målet med intervjuet er å få frem dine personlige opplevelser og erfaringer. Er det noe du ikke skjønner eller ikke ønsker å svare på, er det bare å gi beskjed.

Alle spørsmålene jeg stiller vil også være relatert til din aller første onboarding i din første fulltidsjobb som utvikler.

Intervju

Om intervjuobjektet

Q.1 Hvilket år uteksaminerte du?

Q.2 Hvilket år startet du å jobbe i din første jobb som utvikler?

Om prosjektet

Q.3 Hvor lenge hadde dette prosjektet holdt på da du onboardet?

Q.4 Hvor mange medlemmer var det på teamet da du onboardet?

Q.5 Hvor mange onboardet dette teamet på samme tid som deg?

Q.6 Hva var grunnen til at teamet ditt jobbet remote?

Q.7 Hvilke agile metoder og praksiser brukte teamet?

Om onboarding

Q.8 Kort fortalt: Hvordan var den første tiden som ny på teamet?

i. Hvordan ble du introdusert til organisasjonen, prosjektet og teknologiene?

ii. Hvor mye av dette var fysisk og hvor mye var digitalt?

Q.9 Vet du om teamet valgte å bruke en bestemt strategi på deg under onboardingen? Hvis ja: Hvilken?

Q.10 Hvordan var oppgavene fikk du tildelt i starten?

i. Hvordan var nivået?

ii. Hvordan var beskrivelsene av oppgavene? Var oppgavene veldefinerte?

iii. Hvordan var oppgavenes prioritet og viktighet?

iv. Hva synes du om dette opplegget?

Q.11 Hvordan var bredden på arbeidsoppgavene i starten?

Q.12 Hvordan bidro de andre teammedlemmene til at du skulle bli en del av teamet? Ble det arrangert noen aktiviteter? (Faglig? Sosialt?)

Q.13 Kan du fortelle litt rundt kommunikasjonsflyten i teamet?

Q.14 Hva har i dine øyne vært de største utfordringene? Og «høydepunkter»?

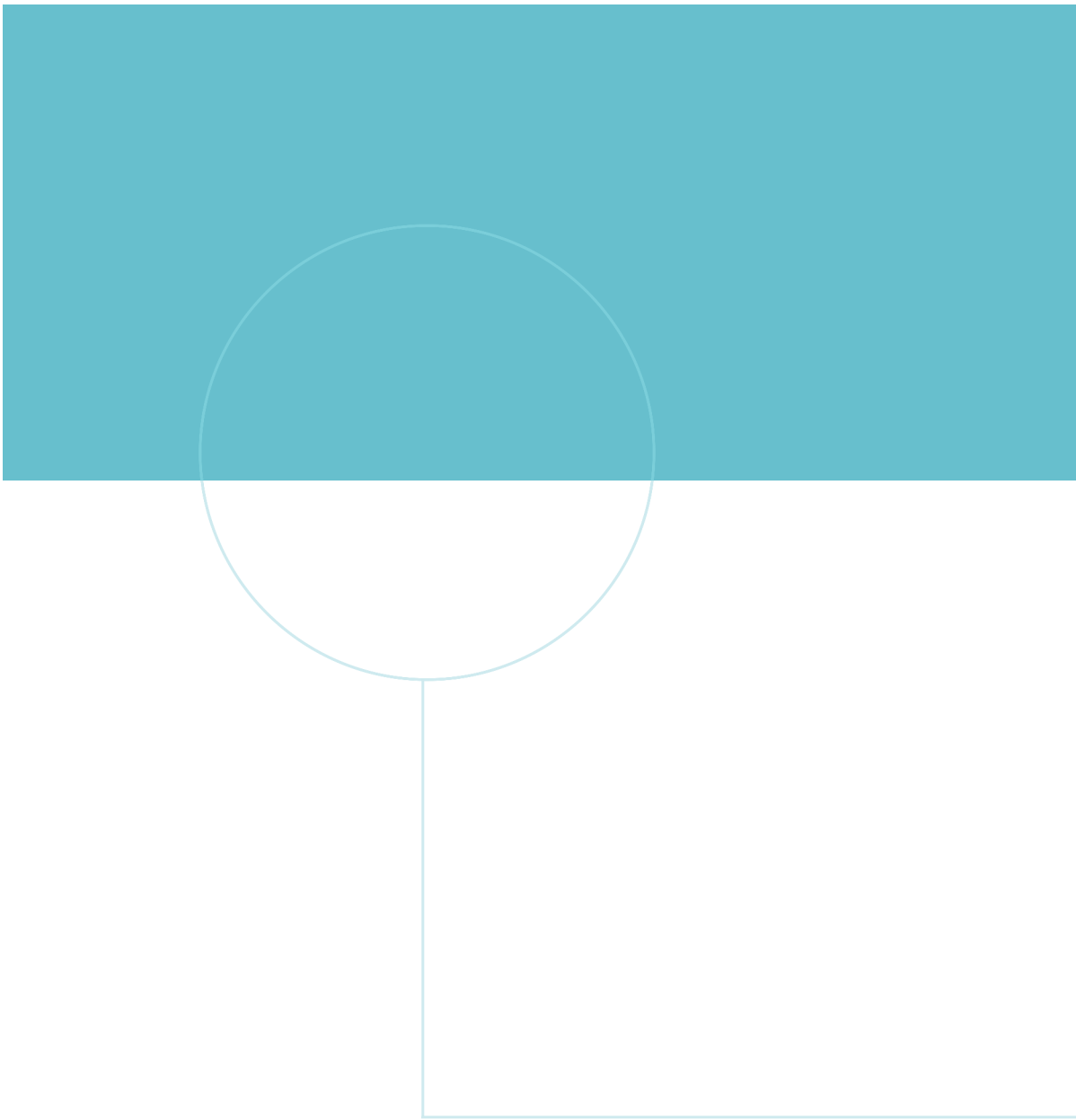
Q.15 Hvor lang tid vil du si det tok før du var ferdig onboardet? Altså at du ga mer verdi til team enn du tok.

Q.16 Hvordan opplevde du onboarding prosessen? Var du fornøyd?

Q.17 Om du kunne bestemt, hva ville du ha gjort annerledes?

Avslutning

Q.18 Er det noe mer du ønsker å tilføye?



NTNU

Norwegian University of
Science and Technology