Agile Software Development: A case study on how agile software development teams adapt to task-based triggers

Navn: Max Trautwein, Anders Aaraas Pedersen

Start: 15.01.2019 09.00

Finish: 01.07.2019 12.00
## Content

**ACKNOWLEDGEMENTS** ................................................................................................. II

**ABSTRACT** ..................................................................................................................... III

**INTRODUCTION** ........................................................................................................... 1
  - Research Question ....................................................................................................... 2
  - Outline of the Thesis .................................................................................................... 3

**THEORETICAL BACKGROUND** .................................................................................. 4
  - Background: Agile Software Development ................................................................. 4
  - Agility in ISD .................................................................................................................. 5
  - Software Development Teams .................................................................................... 7
  - Team Processes and Adaptation .................................................................................. 8

**METHODOLOGY** ........................................................................................................ 12
  - Single-Case Design ...................................................................................................... 12
  - The Case ........................................................................................................................ 13
  - Sampling of Participants .............................................................................................. 13
  - Data Collection ............................................................................................................. 15
  - Transcription and Analysis .......................................................................................... 17
  - Evaluation of Research Methodology .......................................................................... 18
  - Ethical Considerations ................................................................................................. 20

**CASE ANALYSIS AND FINDINGS** ............................................................................. 22
  - Case Description ......................................................................................................... 22
  - Task and Planning ....................................................................................................... 25
  - Task-based Triggers and Team Processes ................................................................. 27

**RESULTS FROM THE CASE ANALYSIS** .................................................................. 31
  - A Model on How Agile SDTs Adapt to Task-Based Triggers ..................................... 31
  - The Components of the Model .................................................................................. 32

**DISCUSSION** ............................................................................................................. 35

**IMPLICATIONS, LIMITATIONS AND FUTURE RESEARCH** ................................. 39
  - Practical and Theoretical Implications ....................................................................... 39
  - Limitations and Future Research ............................................................................... 40

**CONCLUSION** .......................................................................................................... 42

**REFERENCES** ............................................................................................................ 43

**APPENDIX** .................................................................................................................. 49
Acknowledgements

First and foremost, we want to thank our supervisor Jan Terje Karlsen for the guidance, support and feedback he provided during all stages of this research.

We are very thankful for the help from our contact persons at the Nordic financial institution who opened many doors and facilitated a lean-agile cooperation.

We would also like to thank all the participants who provided their time and insights. Without their contributions, this thesis would not have been realizable.

Thanks to Tobias Boisen who helped us with the design of the figures.

Finally, we would like to express our sincere gratitude for the support from our families and friends during this process. They made it possible for us to focus on this master thesis.

Thank you!

Anders Aaraas Pedersen
Oslo, June 2019

Max Paul Trautwein
Oslo, June 2019
Abstract

**Purpose** – The focus in this thesis is on team adaptation from a continuous change perspective. The purpose is to investigate how agile software development teams in a Nordic financial institution adapt to task-based triggers.

**Design/Methodology/Approach** – A single-case study of three software development teams in a product unit reporting to work in agile ways has been conducted. The data is based on 15 interviews, observations and informal talks.

**Findings** – One of the main findings is that the agile software development teams seek to define tasks in a manner that they can be worked on individually, and that it is mainly the adaptation triggers that occur during taskwork which prompts task-related interaction. In addition, the findings support that these adaptation triggers can either prompt small adjustments to taskwork or be more severe and move the focus to evaluation and planning activities before taskwork is continued. Based on the findings, we propose a model describing how the agile software development teams adapt to task-based triggers.

**Research Limitations/Implications** – Future research should use longitudinal research designs to investigate team adaptation from a continuous change perspective in other organizations and contexts, so that these findings can be generalized.

**Practical Implications** – The findings in this research can provide insight in how the agile software development teams continuously adapt, which can assist practitioners in improving the adaptive team performance of their teams.

**Originality/Value** – This research is among the first to study team adaptation of agile software development teams from a continuous change perspective.

**Keywords** Agile software development, Agility, Team processes, Team adaptation, Team microdynamics, Information systems development, Spotify model
Introduction

Approaches to information system development (ISD) that are supposed to improve performance outcomes by promoting agility have become increasingly popular in the last few decades (Campanelli & Parreiras, 2015; Cockburn & Highsmith, 2001; Conforto, Amaral, da Silva, Di Felippo, & Kamikawachi, 2016; Dingsøyr, Nerur, Balijepally, & Moe, 2012). Such approaches are often collectively referred to as “agile software development” and are by many linked to a set of defined methods and practices emphasizing iterative problem-solving, close collaboration and frequent customer interaction (Dybå & Dingsøyr, 2008). Agile software development represents a shift in philosophy that departs from what is considered “traditional” (Nerur & Balijepally, 2007), spurred by the inadequacy of traditional, plan-driven approaches to handle the extensive rate of change in business and technology (Lee & Xia, 2010). The core of agile software development is agility, which most researchers agree concerns software development teams (SDTs) ability to successfully handle change in order to create business value (Conboy, 2009; Conforto et al., 2016; Lee & Xia, 2010; Serrador & Pinto, 2015).

Since agility in ISD involves a crucial aspect of team effectiveness, i.e., the successful adaptation to change (Burke, Stagl, Salas, Pierce, & Kendall, 2006; Mathieu, Maynard, Rapp, & Gilson, 2008), the team processes involved in how SDTs adapt to change are important. Team processes refer to how team members work with each other in order to facilitate goal-oriented work (Marks, Mathieu, & Zaccaro, 2001). Team adaptation can be considered a process involving “adjustments to relevant team processes […] in response to the disruption or trigger giving rise to the need for adaptation” (Maynard, Kennedy, & Sommer, 2015, p. 656). Even though human aspects of ISD have been acknowledged as imperative for a long time (Cockburn & Highsmith, 2001), there are relatively few studies on ISD leveraging the knowledge gained from the social sciences to address such issues (Lenberg, Feldt, & Wallgren, 2015). A literature search revealed that only Kude, Bick, Schmidt, and Heinzl (2014) have leveraged team adaptation theory to better understand agility in ISD. As the team adaptation literature revolves around how teams adapt to changing demands (Maynard et al., 2015), it can contribute to the understanding of how SDTs handle change to create business value.

The team adaptation literature often seems to regard adaptation to novel or disruptive situations (e.g., Burke et al., 2006), thus mainly employing an episodic view on change and subsequent adaptation. The current study seeks to
conceptualize change as both episodic and continuous in nature (Weick & Quinn, 1999). Hence, the current study differs from Kude et al. (2014), who emphasized team adaptation processes from disruptive events. To capture the continuous change, we mainly focus on adaptation to task-based triggers, i.e., adaptation triggers emerging during team members “interactions with tasks, tools, machines and systems” (Bowers, Braun & Morgan, 1997, p. 90, as cited in Marks et al., 2001, p. 357). Adaptation triggers are “those cues that […] can prompt teams to pursue modifications in order to complete their task” (Maynard et al., 2015, p. 653). As this study will illustrate, a definite distinction between the team- and individual level of analysis is misleading for the purpose of this study, which implies that it is appropriate to employ a micro-dynamic approach to team research. This approach emphasizes the “interdependent relations and activities between individuals, and more importantly, […] the organizing aspect of those activities” (Humphrey & Aime, 2014, p. 450). It also entails considering the context in which the SDTs work, even though the focal point of view here are the SDTs themselves.

Research Question

Baard, Rench, and Kozlowski (2014) reported that there is a lack of field studies within the team adaptation literature and empirical studies on team adaptation processes in particular. Furthermore, Maynard et al. (2015) argue that types and the severity of adaptation triggers and the resultant team processes have been largely disregarded in team adaptation theories. A literature search revealed no other studies considering the process of team adaptation with an emphasis on continuous change within the literature on agility in ISD. Thus, the purpose of this study is to employ a micro-dynamic view of teams and explore the practitioners’ experiences with task-based triggers causing the need for the software development teams to adapt and the subsequent adjustments to relevant team processes. The present study has been conducted in a Nordic financial institution, because of the unique access we had to a theoretically interesting case. Hence, the following research question is sought answered: How do agile software development teams in a Nordic financial institution adapt to task-based triggers?

To answer this research question, a case study on three agile SDTs in a product unit that is responsible for the development of one product in a Nordic financial institution has been conducted. When we mention “agile”, we refer to how the product unit’s self-reported ways of working, which we will elaborate upon in
the case description, while “agility” broadly refers to the ability of the SDTs to successfully handle change to create business value. Team adaptation refers to how the SDTs handle change, which implies that we do not focus on the ability to handle change, but the process of adaptation. Even though the case organization reports that their agile ways of working have been quite successful for the SDTs in the product unit, this study does not indicate any level of inherent ability.

The main data collection methods have been observations and interviews, mainly between March to April 2019. Before the main data collection, we had several meetings with the Nordic financial institution in the period September 2018 to February 2019 in order to gain a better understanding of their ways of working and to further develop the research approach. The specific research question and theoretical foundation are the result of this iterative process. By utilizing a single-case design, we sought to get in-depth data from SDTs in the same context. This made it easier to surface the nature of the interactions between the different SDTs in the same product unit and the larger organizational environment. This study is limited to focus on the individual- and team level, however, it will also point towards some implications regarding the larger organizational context.

Outline of the Thesis

First, the theoretical background for the current study will be described. Secondly, we will explain the choice of research methodology and the research process that has been applied. Afterwards, the case will be described followed by the findings of this study. We will thereafter propose a model describing how the SDTs adapt to task-based triggers that is based on the findings. Furthermore, the findings will be discussed based on the model presented. At last, the implications and limitations and directions for future research will be addressed, before the study is concluded.
Theoretical Background

This chapter will review relevant literature for this study. Even though this study builds on an inductive approach, the research question is theory-driven, making it necessary to explain why theory-building is needed (Eisenhardt & Graebner, 2007). We pointed towards some of the reasons why theory-building is necessary in the introduction and will go into further details in this chapter. The choice of literature is based on a continuous research process, where we constantly iterated between the data from the study, the emergent theory and the established theory. Hence, some literature is chosen because of the incremental findings of the study throughout and it links different aspects of the findings to more developed research fields. Iterating between the theory and data in such a manner can be helpful in order to understand the case at hand. Indeed, “rigid adherence to purely deductive or purely inductive strategies seems unnecessarily stultifying.” (Langley, 1999, p. 694). First, a very brief background on agile software development will be presented, followed by a short review of agility in ISD. After that, a definition of software development teams will be provided, followed by the theoretical background on team adaptation.

Background: Agile Software Development

Traditional approaches to software development emphasising detailed up-front plans and extensive documentation have been widely criticised. For example, they are often reported to be unable to meet the users’ needs and requirements (e.g., Lee & Xia, 2010). One of the reasons is that business and technology change with an increased pace which makes it difficult for development teams to identify user requirements and adapt to their changes (Schmidt, Lyttinen, Keil, & Cule, 2001). As a response, the so-called agile methods arose. In 2001, a group of software developers gathered and developed the Agile Manifesto, which describes a number of values and principles of agile software development (Beck et al., 2001). One difference between agile and traditional approaches is that the former prioritise lean processes and dynamic adaptation to change over detailed up-front plans and extensive documentation (Nerur & Balijepally, 2007). In general, agile methods are supposed to deliver software which meets the customers’ requirements in a short time and budget by applying iterative, incremental and adaptive processes (Beck et al., 2001; Brhel, Meth, Maedche, & Werder, 2015).
Researchers keep emphasising that the Agile Manifesto, as well as the methodologies labelled as agile (see Dybå & Dingsøyr, 2008 for an overview of some main methods) were developed mainly by practitioners, and that the approach is anecdotal and lacks a rigorous theoretical foundation or framework (Conboy, 2009; Lee & Xia, 2010; Werder & Maedche, 2018). There is no common definition of what constitutes agility in ISD (Conboy, 2009; Conforto et al., 2016), nor a consistent understanding of why agile software development is supposed to be more effective (e.g., Lee & Xia, 2010). Adopting agile ways of working can be considered fostering an ability to dynamically change the development process as a way of handling rapid change in technology and customer needs (Nerur, Mahapatra, & Mangalaraj, 2005), thus the concept of agility can be considered at the core of agile methods and practices.

Agility in ISD

The current research on agility in ISD has largely been centered around the methods and practices that are labelled agile (Dingsøyr & Lassenius, 2016; Dingsøyr et al., 2012; Dybå & Dingsøyr, 2008) and there has been a trend to judge the agility by compliance to such agile methods and practices (Conboy, 2009; Kude et al., 2014). It has been observed that commercialized methods (e.g., SCRUM or XP) need tailoring to fit specific work contexts (Campanelli & Parreiras, 2015; Cao, Mohan, Xu, & Ramesh, 2009) and that components of certain methods or practices can promote agility in one context but not in another (Conboy, 2009). Thus, an emphasis on certain types of agile methods as what inherently makes software development teams able to more effectively and efficiently handle change can be misleading. As Conforto et al. (2016) point out, agility is not an inherent characteristic of any methods or practices, but comes to life as the team works together to solve their tasks. This implies that agility is a dynamic, temporal and situated phenomenon, what Werder and Maedche (2018, p. 819) refer to as an “emergent phenomenon”, that cannot be fully understood only by the use of methods and practices labelled “agile” (Conboy, 2009).

After reviewing some of the existing definitions, one general distinction seems to emerge. Some conceptualise agility as related to software development methods (e.g., Conboy, 2009), while others conceptualise it in relation to an ability of an entity, for example the software development team (e.g., Lee & Xia, 2010). That is, the distinction mentioned here is not always clear and there is generally
conceptual overlap between the definitions (Conboy & Fitzgerald, 2004; Conforto et al., 2016; Lee & Xia, 2010). The Agile Manifesto, which often is cited to have been very influential in conceptualising agile methods, provides principles and values that qualitatively characterise agile software development, but no proper definition of agility (Qumer & Henderson-Sellers, 2008). Still, its content has been used by some in order to define the phenomenon (e.g., Sheffield & Lemétayer, 2013). One of the consequences of the different ways of conceptualising agility in ISD, can be that research results need careful interpretation, and may appear confusing and conflicting.

Lee and Xia (2010, pp. 90-91) provide an overview of some previous definitions of agility and define it as “the software team’s capability to efficiently and effectively respond to and incorporate user requirement changes during the project life cycle”. Conboy (2009) posits that a software development team may face other changes than just user requirement changes, e.g., related to technology demands (see also Conboy & Fitzgerald, 2004 for examples). According to Conboy (2009), agility in ISD is:

> The continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment (Conboy, 2009, p. 340).

While Conboy (2009) focused on an overarching concept of agility in ISD as a basis for method comparison and development, Werder and Maedche (2018) applied the above definition to software development teams. A difference between these authors can be related to the source of agility. Conboy (2009) seems to consider ISD methods as a source to agility, while Werder and Maedche (2018) consider a team’s self-organization as the main source. This implies that agility can be considered an emergent phenomenon, mainly a result of team processes, and not a static and stable feature of a team based solely on the formal methods it applies in collective work (Werder & Maedche, 2018). Similarly to Werder and Maedche (2018), the mentioned definition from Lee and Xia (2010) implicitly indicates that agility is dependent upon team processes, which relates to how Conforto et al. (2016) stress that it should be considered as tied to an entity. However, these apparent differences may be quite similar in that a self-organizing team develops and/or applies ways of working (ISD methods) that may foster agility. Conboy (2009) considers the term “ISD-methods” to encompass all practices involved in
software development, how they are performed and managed, the order and frequency of the activities, and the inherent values and goals. Thus, the content is implicitly largely people-oriented and points towards that agility is a dynamic multi-level phenomenon.

Agility in ISD relates to the performance of a social entity (SDTs in this case) and how that entity performs in terms of positive adaptation in order to contribute to perceived customer value. This ties agility to adaptive team performance as how it is explained in the team adaptation literature, which can be defined as:

an emergent phenomenon that compiles over time from the unfolding of a recursive cycle whereby one or more team members use their resources to functionally change current cognitive or behavioural goal-directed action or structures to meet expected or unexpected demands. (Burke et al., 2006, p. 1192).

Thus, adaptive team performance can be explained by the patterns of how individual team members contribute to goal-directed action over time (Burke et al., 2006), which ties it to team adaptation processes. As such, performance is here the action itself and not a resultant outcome (Kozlowski & Bell, 2003). While Christian, Christian, Pearsall, and Long (2017) differentiate between routine team performance (completing similar tasks over time) and adaptive team performance (new tasks requiring new actions), this distinction becomes blurry in this case study because of the focus on continuous adaptation. Information systems development is mainly reliant on teams (Capretz, Ahmed, & da Silva, 2017), thus the success of software development highly depends on the performance of the SDTs (Moe, Dingsøyr, & Dybå, 2010). Utilising the knowledge from the more developed team adaptation literature can provide a better understanding of how agile software development teams adapt to task-based triggers and hence contribute to the knowledge on agility in ISD.

Software Development Teams

The software development teams are the entity in focus in this study. However, we do assume that the SDTs are a part of a social work context which must be considered to understand how the SDTs adapt to task-based triggers. Early in the data collection we noticed that in order to understand the team adaptation to task-based triggers, a continuous change perspective was a good fit. Such a perspective emphasises that “organizations are emergent and self-organizing
[which implies that] change is a pattern of endless modifications in work processes and social practice” (Weick & Quinn, 1999, p. 366). Taken to the team-level, a micro-dynamic approach to team research was found most appropriate, as mentioned in the introduction. Here we can define a team as:

(a) two or more individuals who (b) socially interact … (c) possess one or more common goals; (d) are brought together to perform organizationally relevant tasks; exhibit interdependencies with respect to workflow, goals and outcomes; (f) have different roles and responsibilities; and (g) are together embedded in an encompassing organizational system, with boundaries and linkages to the broader system context and task environment (Kozlowski & Ilgen, 2006, p. 79).

As Humphrey and Aime (2014, p. 449) note with regards to team boundaries, the nature of modern teams makes it “increasingly difficult and arbitrary to draw clear boundaries for some team structure and it is equally important for teams to maintain and span those boundaries”. Thus, in essence, it is the “organizing nature of the activities and relationships between individuals that defines the team boundary” (Humphrey & Aime, 2014, p. 450). This puts the focus on the interactions between individual team members, and the self-organizing nature of the teams. It also entails considering multiple levels of analysis, among else, since the self-organizing nature of the teams in real work contexts can hardly be understood properly without taking into account the complex dynamics between levels (Humphrey & Aime, 2014). Since this thesis examines adaptation to task-based triggers, the team tasks are of particular interest in relation to the self-organizing nature of teams emphasised by Humphrey and Aime (2014).

The task represents what a team has to do, e.g., a piece of work and/or an activity, and can function as “a means to prompt interpersonal interaction” (Kozlowski & Ilgen, 2006, p. 80). The task determines how the work is to be done and the coordination requirements that are needed to accomplish the tasks and meet the task demands (Kozlowski & Ilgen, 2006). As Kozlowski and Ilgen (2006, p. 80) state, team processes “are a way to capture coordination of team member effort and factors relevant to it, as well as the alignment of team processes with task demands”. The team processes are the basis for team adaptation theory which will be the topic of the next section.

**Team Processes and Adaptation**

There are different theoretical perspectives on team adaptation (e.g., see Baard et al., 2014 for a review). Process theory is emphasized here since it can help
explain how the SDTs work together to achieve adaptive team performance. It is important to note that the current case study does not seek to uncover any level of adaptive team performance, rather it explains how the SDTs adapt to task-based triggers. Maynard et al. (2015) state that the literature lacks focus on the type of trigger and how different types of triggers can affect the team adaptation process and the outcomes of that process. This thesis concentrates on what Maynard et al. (2015) refer to as task-based triggers. The process theory in this section is based on the temporal framework of team processes presented by Marks et al. (2001).

Team processes depict how teams work together (teamwork) in order to facilitate goal-directed action (taskwork). That is, “the integration of individuals’ efforts toward the accomplishment of a shared goal is the essence of teamwork” (Mathieu, Hollenbeck, van Knippenberg, & Ilgen, 2017, p. 458). Taskwork is essential for team performance outcomes and relies to a large extent on team member competence and team processes (Marks et al., 2001). In this study, the focus is on team processes and not competence. Marks et al. (2001) proposed to differentiate between “team processes” and “emergent states”. Team processes refer to “[team] members’ interdependent acts that convert inputs to outcomes through, cognitive, verbal, and behavioral activities directed toward organizing taskwork to achieve collective goals.” (Marks et al., 2001, p. 357). Emergent states are defined as “constructs that characterize properties of the team that are typically dynamic in nature and vary as a function of team context, inputs, processes, and outcomes” (Marks et al., 2001, p. 357). These describe “cognitive, motivational, and affective states of teams, as opposed to the nature of their member interaction.” (Marks et al., 2001, p. 357). Emergent states come to life through team member action, i.e., things that they do, which “leave an impact on them in terms of influencing their personal and collective psychological states as all this unfolds over time” (Mathieu et al., 2017, p. 458).

Marks et al. (2001) outline three higher order categories, or phases, which depict the general nature of the team processes that are more likely to occur under the given phase. These are “transition”, “action” and “interpersonal processes”. Action phases refer to “periods of time when teams are engaged in acts that contribute directly to goal accomplishment” (Marks et al., 2001, p. 360). Transition phases “are periods of time when teams focus primarily on evaluation and/or planning activities to guide their accomplishment of a team goal or objective” (Marks et al., 2001, p. 360). Interpersonal processes refer to “processes teams use
to manage interpersonal relationships” (Marks et al., 2001, p. 368) and are thought to be apparent both in transition and action phases. The team processes have been related to team performance outcomes, with a tendency of being even more important for performance outcomes if there are higher task interdependencies (LePine, Piccolo, Jackson, Mathieu, & Saul, 2008). Software development in general is typically characterised by such interdependencies (e.g., Strode, Huff, Hope, & Link, 2012), which can attest for the importance of team processes, in line with the common idea that agile software development is fundamentally people-centric (Fontana, Fontana, da Rosa Garbuio, Reinehr, & Malucelli, 2014; Nerur & Balijepally, 2007).

Maynard et al. (2015) built on the team process framework of Marks et al. (2001) when developing a model of team adaptation. They envisioned that the process of adaptation as being “adjustments to certain team processes as dictated by the type of disruption or trigger” (Maynard et al., 2015, p. 659). Based on this assumption, they proposed a model (see Maynard et al., 2015, p. 660). There are other prominent models of team adaptation (e.g., see Burke et al., 2006; Rosen et al., 2011), but the model of Maynard et al. (2015) does not assume a generic, stage-based adaptation process based on a non-specified adaptation trigger, which is why it appeared especially suitable for this thesis. It also addresses concerns about the apparent neglect of different types of adaptation triggers and the subsequent adaptation processes in previous models (Christian et al., 2017; Maynard et al., 2015). As such, it provided some overarching guidelines without constraining this research, as well as pointing towards areas where further theory-building could be helpful.

The model of team adaptation proposed by Maynard et al. (2015) differentiates between two broad categories of adaptation triggers, i.e., task-based and team-based. Whereas task-based triggers are proposed to “prompt teams to adjust their action processes” (Maynard et al., 2015, p. 600), team-based triggers are “apt to prompt teams to adjust their interpersonal processes” (Maynard et al., 2015, p. 660). They also add the concept of severity of the trigger, in which the assumption is that if the trigger is perceived to be severe (e.g., that it represents a novel problem that may have a considerable impact), the team will first adjust transition processes (Maynard et al., 2015). The underlying idea is that triggers and the subsequent adaptation processes are not always the same.
Within the literature on agility in ISD, the case study on three SDTs of Kude et al. (2014) used the team adaptation model of Burke et al. (2006) to study how agile software development teams adapt to non-routine events. They identified three categories of triggers, i.e., technical volatility (e.g., requirement changes), technological disruption (e.g., issues with internal and/or external technological systems) and team instability (stability of team membership). Furthermore, they found three adaptation patterns, whereas they proposed that adaptation triggers which had a large impact led to more extensive adaptation patterns. Kude et al. (2014) thus addressed the type and severity of triggers and subsequent adaptation process. The emphasis of Kude et al. (2014) was on non-routine events and they did not consider the continuous change aspect as this thesis will, nor considered the same team processes. Team process theory can shed light on team adaptation that to some extent has been treated as a “black box” in previous studies within ISD (e.g., Vidgen & Wang, 2009; Werder & Maedche, 2018) and thereby taking the existing literature one step further.
Methodology

In order to answer the research question stated above, an inductive approach of theory-building was chosen in the form of a case study (Eisenhardt & Graebner, 2007). In the following sections, the applied methodology will be described in detail. Firstly, the case will be presented in order to allow the reader to get an understanding of the study’s context. Secondly, the participants as well as the process of sampling will be characterized. This is followed by a description of the data collection and the analysis of the data. Closing, the applied research methodology will be evaluated in order to allow for an assessment of the study’s quality, before ethical aspects will be considered.

Single-Case Design

Yin (2014, p. 16) defined a case study as “an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident”. Case research is also considered as useful when “a phenomenon is broad and complex, when a holistic, in-depth investigation is needed, and when a phenomenon cannot be studied outside the context in which it occurs” (Dubé & Paré, 2003, p. 598). Given the research question, it appears that the software development teams are best studied in their actual environment – the workplace – where they are faced with their usual working conditions and realistic requirement changes. Generally, case studies are well suited for research within the domain of ISD because examining real-life ISD issues enables both practitioners and academia to keep up with the fast-paced changes in the world of ISD (Dubé & Paré, 2003).

Another feature of case studies is the possibility to use multiple data sources in order to paint a richer picture of the object of investigation. This is beneficial since “any finding or conclusion in a case study is likely to be much more convincing and accurate if it is based on several different sources of information” (Dubé & Paré, 2003, p. 615). As will become apparent in the chapter Data Collection, the study at hand made use of observations, interviews and informal talks.

Within the general team adaptation literature it is often stated that future research should employ longitudinal research design, mainly because of the temporal nature of adaptation (Baard et al., 2014). In order to mitigate the drawback of not using a longitudinal design, we sought to facilitate collection of process data.
Langley (1999, p. 692) characterises this type of data referring to van de Ven and Huber (1990) as being “concerned with understanding how things evolve over time and why they evolve in this way”. This is precisely what we were interested in investigating with regards to the research question. Langley (1999, p. 692) further elaborates that “process data therefore consist largely of stories about what happened and who did what when – that is, events, activities, and choices ordered over time”. The interviews focused on extended storytelling with regards to adaptation. Together with the observations, they provided us with process data and enabled us to develop a process theory.

The Case

Basis for the case was a financial institution located in the Nordics. The institution employs several thousand full-time employees and is listed as one of the biggest in its field and region. The informants in this study are members of a product unit within the institution which is responsible for the institution’s main mobile application. This unit consists of around 50 to 60 people, which are mostly co-located on the same floor of the institution’s main office. The “case” in the study at hand are three SDTs within that product unit (see the Case Analysis and Findings for a more detailed description). We selected this case based on its potential theoretical significance, in order to develop theory, which is the main purpose of case studies (Eisenhardt & Graebner, 2007). It is also a unique case in the sense that it represents the institution’s frontier in terms of the future approach to software development. The initial results have been positive enough for the organization that the management considers expanding this approach to other product units. The goal of this study was to investigate the internal processes within these teams, but also the interactions between the teams and with their environment. This would not have been possible if we would have treated the single teams as individual cases, which is why we chose a single case design.

Sampling of Participants

The selection of participants can be described as purposive sampling. This entails choosing participants “in a strategic way, so that those sampled are relevant to the research questions that are being posed” (Bryman & Bell, 2015, p. 429). The main benefit of purposive sampling is that it allows for in-depth study of a phenomenon (Patton, 2015), which is exactly what we intended to do. In order to
answer the research question, a holistic picture of the case was needed. We therefore interviewed employees with different roles and different hierarchical levels. In addition, the distribution of ethnicities and gender within the sample resembles the distribution in the whole unit. The experience of the informants ranged from graduates who joined the institution only a few months ago, to experienced employees who have been working in the institution for over ten years.

The participants in this study were mainly chosen by two superior members of the unit. We only decided which roles the participants should have but did not choose the participants themselves. Each of the 15 participants was interviewed once, leading to a number of 15 interviews. See the following table for an overview of the interviews:\(^1\)

**Table 1: Overview of interviews**

<table>
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>Role</th>
<th>Main Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15.03.2019</td>
<td>Agile Coach</td>
<td>- Development processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Agility in the Nordic financial institution</td>
</tr>
<tr>
<td>2</td>
<td>19.03.2019</td>
<td>Product Owner</td>
<td>- Orientation towards the market</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Structure of the product unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Roles</td>
</tr>
<tr>
<td>3</td>
<td>21.03.2019</td>
<td>Frontend Developer</td>
<td>- Structure of the product unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Team adaptation processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Team memberships</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Decision processes</td>
</tr>
<tr>
<td>4</td>
<td>21.03.2019</td>
<td>Backend Developer</td>
<td>- Structure of the product unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Team adaptation processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Trial and error</td>
</tr>
<tr>
<td>5</td>
<td>03.04.2019</td>
<td>Frontend Developer</td>
<td>- Roles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Technical freedom</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Team adaptation processes</td>
</tr>
<tr>
<td>6</td>
<td>03.04.2019</td>
<td>UX Designer</td>
<td>- Structure and development of the product unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Roles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- The product</td>
</tr>
<tr>
<td>7</td>
<td>03.04.2019</td>
<td>Product Manager</td>
<td>- Roles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Structure and flexibility of the product unit</td>
</tr>
<tr>
<td>8</td>
<td>05.04.2019</td>
<td>Backend Developer</td>
<td>- Structure and development of the product unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Team adaptation processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Interdependencies</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Tools</td>
</tr>
<tr>
<td>9</td>
<td>08.04.2019</td>
<td>Backend Developer</td>
<td>- Structure of the product unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- The product</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Roles</td>
</tr>
<tr>
<td>10</td>
<td>08.04.2019</td>
<td>Backend Developer</td>
<td>- Structure and development of the product unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Roles</td>
</tr>
</tbody>
</table>

\(^1\) All 15 interviews revolved around team adaptation processes. In order to make them more distinguishable, the column “Main topics” was added. However, the topic “team adaptation processes” is not mentioned in every row, thereby making space for more unique topics.
Before the actual collection of data started, we had several discussions and coordination meetings with different units and departments of the institution. We did this in order to determine which team or unit would be most suitable for investigating the research question. This “pre-study” lasted for the period of September 2018 to February 2019. After the product unit and the SDTs within that unit were chosen to be the unit of analysis, we observed a subset of those teams on multiple days during two weeks in the beginning of March 2019. These observations were followed by a series of interviews from mid-March 2019 to mid-April 2019. In addition, informal talks with the participants further contributed to a deeper understanding of the processes happening in the unit. These different types of data as well as the fact that employees with different roles and from different hierarchical levels were interviewed are generally referred to as “data triangulation” (see for example Bryman & Bell, 2015; Patton, 2015; Yin, 2014). The data stems from different sources, however all data contributes to investigating the same phenomenon (how SDTs handle task-based triggers).

We designed both a common method for logging the observations as well as a semi-structured interview guide (see appendix A). This allowed every observation to follow similar procedures and every interview to follow similar case study questions. We informed the participants on beforehand about the purpose and circumstances of this research, and about what a participation entails for them. The degree of information was sufficient in order to allow the participants to make an informed decision about whether to participate in the observation and interviews or not. However, we did not want to prime the informants with popular terms related
to the research question, which is why we did not mention terms like “agility” or “agile” before the interviews. This decision was taken because people seem to have many different interpretations and understandings of the concept of agility and we wanted to prevent the participants from thinking about agility (or even worse, using the internet for some inspiration) before the interview, so that the participants had to answer freely and unbiased of others’ understandings of the concept.

The observations were done before the interviews, which enabled us to get an idea of the participants’ work processes and procedures. This also allowed us to develop a meaningful interview guide. We observed a subset of the SDTs, of which all members were located in the same area of the same floor. During the observations we sat in a corner of that area where we were not very visible and quietly observed, in order to not distract or affect the developers. In addition, we were informed in the course of the observations that we could attend several coordination meetings. This “freedom to make adjustments during the data collection process” (Dubé & Paré, 2003, p. 618) is another aspect of case studies which makes this form of research very suitable for this thesis. We logged the observations in a common way according to the previously developed method. This entailed that we used time stamps for all observations, we noted everything that we observed without interpreting or judging what was happening, and we made use of a number of behavioural markers, inspired by Rosen et al. (2011).

The interviews were executed in a semi-structured manner. The use of the interview guide ensured that all important topics were covered. However, a large degree of flexibility was necessary in order to allow us to be responsive to the participants’ statements (Bryman & Bell, 2015). Furthermore, the complex nature of the topic as well as the fact that everyone has a different understanding and perception of certain terms or questions made it necessary to have enough freedom while conducting the interviews. The questions in the interview guide were designed in order to gain “sequential” (however not longitudinal) data which enables us to tell consistent stories (Langley, 1999).

The interviews were conducted at the institution’s main office and had a duration of 30 to 45 minutes each. They were recorded after the participants gave their consent to do so, in order to enable a later transcription and analysis of all details that might not be captured by taking notes (Bryman & Bell, 2015; Yin, 2014). All participants were given the freedom to have the interview conducted in either English (which is the product unit’s working-language) or Norwegian (which
is the mother language of the majority of the informants), in order to enable them to speak as freely as possible. As Polkinghorne (2005) pointed out, the use of native language enables informants to enrichen their answers for example with regards to the use of metaphors and expressions (see also Bryman & Bell, 2015). In addition, the interviews were conducted at the company’s site in the group rooms which the employees often use for meetings, so that the environment was common to the employees and the interview was not interrupted. This contributed to shaping an atmosphere in which the participants could speak freely (Bryman & Bell, 2015).

The sample size was 15, which appeared to be a sufficient number of interviews, since we experienced what is known as “theoretical saturation” (Bryman & Bell, 2015). This concept states that one has reached a sufficient number of interviewees when an additional interview does not contain vital new insights or dimensions.

**Transcription and Analysis**

In order to allow for an accurate understanding of and quotation from the data, the interviews were transcribed in the language the interview was conducted in. In this way there was no loss or bias of data by translation (Bryman & Bell, 2015; Polkinghorne, 2005).

The analysis followed multiple strategies proposed by Langley (1999). At first, an open coding process was executed. We read through the data multiple times and formulated categories which summarized certain bits of data. This was done by writing both notes and the names of the categories on the margins of the printed interviews. In addition, each relevant quote was marked with a code, which allowed only us to trace back quotes to individuals, but no one else. We then cut out the relevant quotes and sorting them according to the categories. This process resulted in having 14 categories. Each of them was cross-checked between both of us in order to prevent personal bias affecting the study (Yin, 2014). After the data was grouped in categories, the initial coding was reviewed, and the data within the categories was distilled into the quintessence. Afterwards, we applied what Langley (1999) called “Visual Mapping Strategy” as a form of axial coding. We identified similarities, differences and connections between the categories by visualising them on a whiteboard. This enabled us to identify three core categories, related to team adaptation to task-based triggers, namely *task and planning, task-based triggers and team processes* and *enablers and constraints*. The data will be presented based
on these three categories in the chapter *Case Analysis and Findings*. Then, we further analysed the content of these three categories and the underlying stories in order to identify the patterns that explain how the SDTs adapt to task-based triggers. These patterns were visualised in graphical form, which after several revisions and discussions resulted in the proposed model (see figure 3).

**Evaluation of Research Methodology**

It is important to identify potential sources of errors within a study’s research methodology, in order to evaluate the quality of research. Especially in qualitative research, trustworthiness is difficult to establish. The reason for this might be the notion that in naturalistic research, criteria as reliability and validity cannot be evaluated as easily as in positivistic work. In order to approach this problem, Guba and Lincoln (1982) developed four criteria in order to support qualitative researchers in evaluating the quality and trustworthiness of their research: credibility, transferability, dependability and confirmability. In this section these criteria will be described and applied to the study at hand.

When considering the *credibility* of a study, Guba and Lincoln (1982, p. 246) pose the question “Do the data sources (most often humans) find the inquirer’s analysis, formulation, and interpretations to be credible (believable)?”. What arguable contributed most to this criterion is the fact that we presented the analysis and findings to the financial institution before submitting this thesis. The institution confirmed that the interpretations were comprehensible and realistic. Moreover, the applied triangulation, i.e., use of several data sources in this study make the interpretation of the data more genuine (Guba & Lincoln, 1982). The fact that the interpretations from the interviews are in line with the “slice-of-life” observations done previously further confirm the study’s credibility (Guba & Lincoln, 1982, p. 247). Having done observations of the team before developing the interview guide enabled us to ask meaningful questions. In addition, a pilot interview was conducted which served as a test of the questions and phrasings. Furthermore, the participants were given the possibility to withdraw their consent to participation at any time as well as to check quotations which were used in the final version of this thesis. Shenton (2004) names this as another way of contributing to credibility.

Guba and Lincoln (1982) put the criterion *transferability* on a level with generalizability, meaning that for a study to fulfil the criterion of transferability, the
sample should in some way be representative of its population. Like that, the study will be (at least to a certain degree) applicable to other contexts and settings. Typically, qualitative research is not known for a high degree of generalizability. Nevertheless, Guba and Lincoln (1982) name randomisation and stratification of the sample as possible strategies, in order to still contribute to the described criterion. Stratification was unfortunately not realisable in this study, but the sample was selected in a somewhat random way, seeing as it was not us who chose the participants, but a few product managers in the product unit.² In addition, the sample covers most of the important roles with regards to the SDTs which are represented in the unit.

Another recommendation of Guba and Lincoln (1982, p. 248) related to the sampling is to make use of “theoretical/purposive sampling”. As explained in the chapter Sampling of Participants, the sampling was done with the goal to include a broad spectrum of participants. In addition, several authors mention that another means to contributing to transferability is a “thick description” (Guba & Lincoln, 1982, p. 248; Shenton, 2004, p. 70) of the case and context the data is taken from. We tried to meet this demand by describing the organisation, product unit, its context and the participants of the study.

The criterion dependability is compared to reliability and replicability by Guba and Lincoln (1982). The goal of this criterion is that a different researcher would reach similar results if he would repeat the study under the same conditions, with the same participants and methods but at a different place and time. This can in practice be reached by reporting the research processes in detail as well as handling and processing the data in a precise manner (Golafshani, 2003; Shenton, 2004). This chapter aims for exactly this detailed description of the research process. In addition, the recording of the interviews as well as the absence of potential bias by translation allowed for a precise transcription and eventually precise case study data. Another strategy which contributes to dependability is the use of “overlap methods” (Guba & Lincoln, 1982, p. 284). The use of multiple methods in this case study has been described in the chapter Data Collection. Furthermore, the fact that both of us constantly questioned each other’s work and agreed on common interpretations (for example by cross-checking the categories in the analysis of the data) contributed to inter-rater reliability.

² This also helps preventing selection bias.
Guba and Lincoln (1982) describe their last criterion *confirmability* as the qualitative equivalent to objectivity. Shenton (2004) further defines this criterion by saying that in order to reach confirmability, it has to be made sure that the content of the data stems from the participants themselves and is expressed in a free manner, instead of being affected by the researchers’ expectations or even preferences. Both the described use of multiple data sources as well as the randomisation of the sample at the study at hand contribute to confirmability by reducing the risk of investigator bias (Guba & Lincoln, 1982; Shenton, 2004). Guba and Lincoln (1982, p. 248) suggest another strategy in order to ensure confirmability, which they call “confirmability audit”. Following this approach, the researchers should make sure that they record both the data and the path of analysis in a way that allows them to trace every finding through the different steps of analysis to back to original data. In this study, this was ensured because we assigned specific codes to every quote of the participants and kept track of which quote contributed to which category. In that way, we were able to accurately trace which quotes every finding was based on.

**Ethical Considerations**

Diener and Crandall (1978) define four major issues of research ethics: harm to participants, lack of informed consent, invasion of privacy and deception. In this chapter, the study’s compliance with these ethical issues will be evaluated.

There are different forms of *harms of participants*: physical harm, mental harm such as stress or harm to future career possibilities of the participants (Bryman & Bell, 2015). *Physical harm* can be considered irrelevant in this study, because it did not require any physical demanding interactions. It can be assumed that *mental harm* did not occur either. We tried to design the participation of the informants as convenient as possible: the informants could choose a date and time for the interview which suited their calendars and workloads, the interviews were conducted at the participants’ office building which they are used to and the participants could choose between having the interview conducted in English or in Norwegian. All that contributed to that the interviews should not have been too stressful for the participants. Lastly, the *future career possibilities* of the informants were not affected neither, since the participation in the study was voluntary and the contributions were given anonymously.
Informed consent means that “prospective research participants should be given as much information as might be needed to make an informed decision about whether or not they wish to participate [in the study]” (Bryman & Bell, 2015, p. 139). The participants did receive an information sheet with all important information about the cause and process of the study, the handling of data, the participants’ right to withdraw from the study at any given point of time as well as relevant contact details. In addition, we summarized the content of that information sheet to every participant before the interview started. The informants therefore had enough information in order to give their consent “freely […] informed, and in an explicit form” (The Norwegian National Research Ethics Committees, 2016). The consent was given either in written form or in recorded verbal form.

The Social Research Association (2003, p. 27) states regarding invasion of privacy that participants “may feel they have been treated as objects of measurement without respect for their individual values and sense of privacy”, referring to Cassell (1982). In the study at hand, the whole process of data collection was designed in a way which respects the participants as individuals. Especially the semi-structured way of conducting the interviews allowed the participants to tell the story from their point of view. The Social Research Association (2003, p. 27) further mentions that the participants’ privacy often is invaded “by collecting ‘too much’ information”. This has been avoided by repeating multiple times that the participants have the right to not answer any question without stating any reason. The fact that no informant made use of this right further proves that invasion of privacy was not an issue in this study.

Bryman and Bell (2015, p. 144) describe the fourth criteria of ethical consideration, deception, as “when researchers present their research as something other than what it is”. This is especially relevant in qualitative research, since interpretation of data naturally plays a much larger role than in quantitative research. In order to make sure that the report at hand represents what the data says, we had a meeting with the financial institution in which we presented the findings and doublechecked that no misinterpretations were done, before submitting this thesis.

Closing it should be mentioned that this research project was registered with and approved by the Norwegian Centre for Research Data (NSD) in March 2019. NSD has high demands with regards to meeting ethical standards.
Case Analysis and Findings

In the following chapters, the case description and the findings which are the result of the analytical process outlined in the chapter *Transcription and Analysis* will be presented. At first, the setup of the product unit will be described in more detail, because it is the background and context of the main findings. Afterwards, the findings will be presented in two categories: *task and planning* and *task-based triggers and team processes*. The findings related to *enablers and constraints* will be included in the other two chapters, because the identified enablers and constraints surface continuously in relation to the other two categories. During this chapter, quotes from the interviews will be used in order to support the description of the original data and to make sure that the description is not merely a product of interpretation.

Case Description

The product unit started seeking to work in a more agile way during 2017. They took inspiration from the so-called “Spotify Model” (see Kniberg & Ivarsson, 2012) and adjusted that to their own situation and needs. As one product owner described: “We took some inspiration from Spotify, and then we have tuned the model exclusively for us” (Product Owner). The period of adjustment was characterized by a lot of “trial and error”, as several informants told us. From autumn 2018 on, the informants felt that the trial and error period was mostly overcome. They reported to perceive the setup meanwhile as relatively stable. However, one aspect of their setup is that it is very fluid and not fixed, since they are continuously seeking to improve it. Therefore, the following description of the structure should be seen as a snapshot, rather than a finalised state. At the time of this study, the product unit had already released a product to the market and was working on developing new features and continuously improving the product.

The whole product unit is called “tribe” and consists of three “squads”, which all contribute to the same product (the institution’s main mobile application). Those squads are medium sized teams which consist of around eight to twelve

---

3 The used terms (like “squad” and “tribe”) are taken from the way Kniberg and Ivarsson (2012) describe the setup of Spotify. The institution in the case at hand developed their own terminology in order to underline that they have adjusted Spotify’s model to their own situation and needs and in order to increase identification and commitment of the members of the unit. The exact terms are not disclosed for reason of anonymisation.
people. Each squad has a common topic and works with tasks related to that topic. Some of these tasks are easy to assign to a certain squad, while other tasks are related to or have an impact on several squads. Each squad consists of people with different roles, meaning that each squad is able to develop IT solutions on their own, from frontend\(^4\) to backend. In addition, people with the same roles (e.g. frontend-developers) are connected in so-called “chapters”, which range across the different squads. Like that, the developers can discuss their skill-specific issues with other people who have the same role and therefore a similar expertise, even though they are not in the same squad. For example, when an Android or iOS related problem which affects multiple squads occurs, that problem is handled by the respective chapter. See figure 1 for an overview of the squads and chapters.

![Figure 1: Squads and Chapters, based on Kniberg & Ivarsson, 2012.](image)

As can be seen from figure 1, each squad is headed by a product owner (PO). Their function is to enable their squad to focus on the technical development by taking care of administrative and coordinative tasks. They are in frequent contact with the developers in their squad. A main instrument for this is regular meetings: each squad has a daily stand-up and a weekly meeting. During the daily stand-ups, the developers update the others in the squad about what they have been working on the day before, which issues they encountered, how they solved these issues and what they will be working on for the rest of the day. This is done in order to create

\[^4\) For reasons of simplicity, the term “frontend-developer” refers to both Android and iOS developers in this report.\]
a mutual understanding of what the team is working on. Some developers reported that these meetings help anticipating which tasks might come in the near future or which changes need to be done in the existing code. The purpose of the weekly meetings is to get an overview of the current progress, address coordination issues and set the priorities in terms of tasks for the upcoming week. Furthermore, these meetings align the members of the squad and create a common overview of what the squad is trying to achieve and how the single tasks contribute to the big picture.

In addition to facilitating regular communication within their squads, the product owners are also the link between the two different parts in which the unit is (at least “on paper”) split in: the IT side (i.e., the members of the product unit who are doing the technical work) and the business side. It should be mentioned that this split exists mainly for organizational reasons in terms of who the individuals report to. In practice, the difference is less visible and both sides collaborate closely on both business and technical tasks. Both sides are governed by one person (a so-called “section head” for the business side and a “co-pilot” for the IT side). While the product owners report directly to the co-pilot, they also are in close contact with the product managers, which are their equivalent on the business side. The product managers report to the section head and are supposed to develop the long-run roadmap or vision. Furthermore, their task is to ensure that the squads follow this roadmap. This is done by attendance of the weekly meetings as well as by frequent interaction with the product owners, who are responsible for breaking down the long-term goals and developing short-term actions. See figure 2 for an overview of the unit’s two sides.

![Figure 2: The two sides of the product unit.](image)

---

5 SH refers to section head, PM refers to product manager, CP refers to co-pilot and PO refers to product owner.
The developers in the product unit can be divided into two categories, based on their responsibilities. One of them is the frontend, which deals with the application’s “surface” and how the customers interact with the application. On the other end, the product unit draws on a lot of data from other departments and systems in the institution. This leads to the emergence of manifold interdependencies. The product unit’s backend developers cooperate and coordinate with these other departments in order to receive the needed data.

**Task and Planning**

The data revealed that tasks can stem from different sources. The informants reported that a typical example for where a task can come from is that the business side (for example the product managers) comes up with a new feature that should be integrated in the mobile application. What is required for that is that the business side is “open-minded” (Product Owner) and continuously observes the market, in order to identify external market opportunities or “technology shifts” (Product Owner), because these “developments are happening at extremely rapid rates” (Section Head). In addition, the informants stated that there are many legal changes within the financial sector, which means that established rules “open up and you need to respond quickly” (Section Head). Another source of change is when the developers identify a “technical opportunity that [the product unit] could realise quite fast” (Section Head). This source may lead to “smaller” improvements or existing features which might be easier to implement.

All participants reported that irrespective of the source, the new goal gets broken down into approachable tasks. These tasks are often complex, because they are difficult to analyse. The informants stated several reasons for that. One of them is the “enormous amounts of dependencies” (Backend Developer) with other parts and systems of the institution, which are outside of the unit’s control. Some of these interdependencies cannot be foreseen before starting the actual development. They online arise during the coding itself, as one developer explained:

> At first, things can seem very easy. […] But it is a lot of ordering small pieces that has to be in place, and then you kind of realise how the production line actually goes. That the backend has to be in place for you to write the code correctly (Frontend Developer).

In addition to this, many informants stated that the used technical systems may fail, and the prioritisation of tasks might suddenly change. The combination of these factors was reported by the informants to make the length of the tasks almost
impossible to estimate. “If you are an experienced software developer you get a little better in assuming [how long a task may take]” (Backend Developer), but they are still not able to assess the duration of a task on beforehand with certainty. One approach towards this issue which was emphasised by most developers is to plan less and to start the work in order to reveal more task-relevant information that is needed, as described by this informant: “It seemed like we had everything that we needed to have and we didn’t realise until we made it, that we actually needed a resource from an external system” (Frontend Developer).

The informants named several ways of continuously striving towards reducing the described interdependencies. Bigger tasks get broken down into smaller bits, which is supposed to reduce interdependency. The analysis showed that a complex product feature might for example require data from several other parts or systems of the institution. After breaking down this feature into small tasks, one such task might be depending on only one other system, which is easier to handle. A similar approach would be to split up tasks in a way that the interdependencies are kept within the same squad. As one of the informants described:

We try to define the boundaries and we try segregate the work as far as possible. So, if we have different boundaries, then we don’t have interdependent work. So, then it’s fine I believe, because then you don’t have any interdependency (Backend Developer).

One reason for this is that the members of one squad know each other’s skills as well as schedules better than those of other squads. As one developer explains: “So inside the squad it’s easier to know what everybody does, and I can say ‘You need to do that now, for me. We need to fix that now’. But I have no idea what people in a different squad are doing” (Frontend Developer). In addition, the informants stated that it was easier for them to approach a colleague who they have a social connection to, than someone who they have never talked to. Having social activities was reported to help building this connection as well as getting an overview of which people have which knowledge or skills in the whole product unit. As one developer explained after we asked whether social gatherings help when they face an issue later on:

Yeah, because of the social gatherings I know who to go to. Or at least where to start to figure out my question. Yeah it is just easier to talk to them as well. It is not as scary when you have had the first “hello” (Frontend Developer).
Another way of how the product unit seeks to handle the interdependencies that affect the SDTs is entailed in certain roles. Product managers and owners described that they deal with the entities which are external to their squad and which the squad is reliant on. They do the administrative and coordinative work and, in that way, “try to shield the developers so that they can actually have to time to focus” (Product Manager) on their core job: developing. In addition, the backend developers also reported that their role entails cooperating with parts of the institution which the product unit draws its data from. The nature of their roles makes them deal with these dependencies during the developers’ taskwork as well.

The informants described that the process of breaking down a problem into bits results in “approachable” tasks which are stored in the backlog. The developers can later pick tasks from this backlog. The developers described that before a new task is started, they often talk to those parts of the unit on which the task is dependent on and which are affected by the task, “so we are in alignment” (Backend Developer). By discussing and planning the developers reported that they try also to map out a possible outline for how to solve the task. In addition, they seek to find out which resources they need in order to solve the task, and where to get these resources from. Initially, a task is supposed to be kept within a squad, but as one of the product owners explains, “if there is something that we need to accomplish within a specific timeline, we work very closely [with people from other squads]” (Product Owner).

**Task-based Triggers and Team Processes**

After having done some initial planning, the developers start with their actual coding work. The data reveals that the developers have a rather large amount of flexibility when it comes to how to solve a task from a technical perspective. As one of the product owners describes: “With respect to coding and which is the best technical decision to actually follow, all of that are done by the developers themselves” (Product Owner). However, they have to stick to certain boundaries, which have been defined by for example the project owners or managers. One example of these boundaries is the priorities, as one of the developers describes: “We go with the priority given by the product owners” (Backend Developer). Still,

---

6 Here, backlog refers to an overview of uncompleted and on-going work, organized with information like priority, stage in the development process and who that are assigned or works with particular tasks that is available to all team members.
solving the problem has a higher priority than following a plan. This can be seen in the way one of the frontend developers described the flexibility of working: “We don’t have a detailed plan how everything should be, we have a problem that we are going to solve” (Frontend Developer).

The developers reported to be by default working “mostly alone” (Frontend Developer), meaning individually. The data revealed that when certain triggers arise, the developers ask for support from one or more colleagues. This can for example happen when a developer encounters a coding-related issue which “I [a developer] can’t figure out on my own” (Frontend Developer). The majority of informants described that this trigger usually does not require more than a relatively short task-related discussion with a colleague or coordinating work to be solved. One developer described the continuation of a task after such an interruption as easy and “a matter of minutes” (Frontend Developer).

Criteria about who the developers interact with when issues arise were stated to be expertise and availability. On the question what would determine which people a developer would talk to in such a situation, he answered: “Who has time and who has competence” (Frontend Developer). It was stated that it also plays a role how good the two people know each other, and that they have met before, because “then it is much more easy to talk to them later” (Frontend Developer).

The informants reported that being co-located in general was contributing to having beneficial task-related interactions. Being located on the same floor as other developers did not necessarily make it easier to approach others, without feeling like a part of the team. One of the informants described:

We were just in the corner over there [the same room as the others]. But still it was weird, because it was still like this distance and you didn’t feel like a part of the team, since you were in your own section. Kind of strange. Because then it was not so natural to join other conversations. Too far away (Frontend Developer).

There are several sources of feedback which the developers reported to receive while doing taskwork, and which is another form of trigger. Feedback is for example generated by testing the produced code in a testing environment: “According to the feedback from [the test environment] we redesign or fix something” (Frontend Developer). A second source of feedback can be a designated
team member who reviews the produced code. As one developer puts it: “Sometimes I change things in other peoples’ codes, and then I need them to review my change, which can lead to discussion” (Backend Developer).

The data revealed that other triggers have a bigger impact on the developers. The majority of developers described that some triggers lead to that the fulfilment of a task is blocked. One example for this are meetings, which were reported to be taking the attention away from taskwork temporarily. Meetings have been described as having an interruptive character, because “it means that [the developers] cannot really develop” (Backend Developer). Returning from a meeting to a task can take up to “thirty, forty-five minutes to get into [the task] before you can get productive” (Backend Developer).

The data showed that another source for being blocked is typically a failure of technical systems that the developers are reliant upon, e.g., the development tools or the test and integration environments. “Sometimes the login doesn’t work for the apps in the test environment” (Backend Developer). Another reason for why a task can be blocked are dependencies on other systems and parts in the product unit or in the rest of the institution. A problem which has been reported frequently is that those other parts in the institution are perceived as not having an understanding of how and why their contribution is important to the product unit and therefore to the customer. As one product owner describes:

> We are lacking an understanding of people who work further down [referring to other parts of the institution which are external to the product unit], they do not necessarily understand what they are actually delivering. And to whom. So, they are lacking the view for the customer and the importance of what they are doing (Product Owner).

Another issue related to this is that the external parts do not respond to issues raised by developers in the product unit with sufficient pace. One product owner stated that “they do not work equally fast” (Product Owner).

The developers reported that when the blockage occurs in the beginning of the development process, “you are in a phase where you can bypass the problem” (Backend Developer). Some tasks, coding wise, can be worked on and finished while dependent systems are out of function. Another approach to a blockage is to pause that task and start working on another one, as one developer describes: “So it

---

7 A type of quality assurance, as well as source of feedback, where another team member evaluates the work (the written code) of the task-doer.
is not like we are not doing anything at that point in time, we just keep it to the side and move on to another task” (Backend Developer).

When the developers are in a later phase of the development process, it is usually not possible to bypass a blockage. As one developer described: “When you are done with your development and you are probably proceeding to an integration, testing kind of a thing, then you need all your services to be working” (Backend Developer). If the systems fail at this point of time, “there is no other way, you are kind of stuck. So, your deployment is delayed, your testing is delayed” (Backend Developer), meaning that the developers can still start working on another task, but the delivery of the feature which is affected by the blockage will most likely not be within the time plan.

The developers reported that another example for a trigger which has a big impact on them is when the business side changes the priorities of tasks or features. One developer exemplified: “Every now and then I work on something and then someone comes and says, this isn’t prioritised any longer, do this thing instead” (Backend Developer). The business side then decides which features are (more) important to the application and should therefore be worked on prior to other features. “One way the features are prioritised is that we take feedback from the customers into account” (Backend Developer). A typical source of such feedback is that “customers call the customer centre” (Product Owner). One example from the data for the business side changing priorities is that “we [the developers] worked on one feature and suddenly the business just bought another companies solution” (Frontend Developer). This led to that the developers’ task suddenly was irrelevant and they had to start working on something else.
Results from the Case Analysis

Based on the analysis and findings outlined in the previous section, we propose a model (figure 3) that describes the main pattern of how the agile software development teams in this case study adapt to task-based triggers. We will here outline the model and its components.

A Model on how Agile SDTs Adapt to Task-Based Triggers

Figure 3 visualises the findings and the answer to the research question in a way that captures the general patterns and the variation in the findings. The analysis revealed that previous theory could provide a useful organizing framework for the findings, thus we have used terms derived from the team adaptation and team process literature to explain key parts of the model. For example, the model is in line with the framework of Marks et al. (2001), as in that task accomplishment starts with a transition phase, before an action phase. Even though these phases in reality may frequently blend into each other, the findings support the general idea that there are times where one phase is more salient than the other.

One of the core assumptions of the model is that the organizing nature of the teamwork revolves around the tasks. This implies that the micro-dynamic view of teams is a foundation of the model. Since it is based on a task-cycle, it entails that different team members can be at different places in the model at the same time.
The task can be considered the natural starting point of figure 3. The model contains six main parts, which are described in table 2.

### Table 2: Description of the components in figure 3

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>A piece of work or an activity that prompts team members to interact.</td>
<td>It comes with various degree of definition, which means that team members may define tasks further before planning.</td>
</tr>
<tr>
<td>Planning</td>
<td>Team members’ preparation for taskwork, either alone or together with other team members.</td>
<td>Even if done alone, the task a developer works on contributes to the application as a whole.</td>
</tr>
<tr>
<td>Taskwork</td>
<td>Team members interactions with tasks, tools, machines and systems.</td>
<td>Goal-directed action.</td>
</tr>
<tr>
<td>Trigger</td>
<td>Cues emerging from taskwork or other sources (feedback) that can prompt teams to pursue modifications in order to complete their task.</td>
<td>Task-based triggers. These triggers can have different levels of severity.</td>
</tr>
<tr>
<td>Adjustment</td>
<td>Team processes, like back-up behaviour or other coordinative behaviors, which results in smaller changes related to taskwork.</td>
<td>Does not move the main activities or attention from goal-directed action.</td>
</tr>
<tr>
<td>Evaluate &amp; Adapt</td>
<td>Assess situation either together or alone and adapt the task and/or planning.</td>
<td>When a trigger results in moving the main activities or attention towards transition processes.</td>
</tr>
</tbody>
</table>

### The Components of the Model

*Task* represents a piece of work that team members are supposed to address. It can enter a team through the backlog, be directly assigned to team members from others (typically managers) or be created by team members themselves. The task does not represent an objective amount of work, but an interpretation of what needs to be done in order to reach a defined outcome. The findings revealed that a common feature of the task category is that its complexity often makes it difficult to plan the amount of work and/or resources needed to finish the task, which can make initial time estimates inaccurate. The main source of this complexity stems from the known and unknown interdependencies with other systems that are outside of the SDTs’ control. In addition, the task can have different levels of urgency and impact, which can result in a more elaborate transition phase, where many team members are involved.

*Planning* represents team member preparation for taskwork, which includes an assessment of what is needed in order to tackle the task and preparation of the tools and/or machines used in taskwork. The case analysis supports that if the task is perceived to be urgent and/or impacts multiple other team members, or other
departments in the organisation, more people are involved in planning activities to further define tasks and organize the work are conducted. Such activities are represented by the arrows between “Planning”, “Evaluate & Adapt” and “Task”. The findings suggest that tasks are sought to be defined so that that one team member can work on it as independently as possible from both other team members and other systems. Additionally, if it is a feature-related task, it is often the squad that handles it internally, if it is a task related to a specific technology (e.g., iOS), then it is handled by the respective chapter. When a team member starts planning for taskwork, that developer interacts with other team members, mainly based on the perceived expertise of others and who is available. In addition, having a social connection to the others was mentioned as lowering the threshold for approaching the other team members. The findings suggest that the task itself can function as an adaptation trigger, which makes the SDTs adapt their response, similar to the team adaptation model of Burke et al. (2006).

*Taskwork* represents what the team members do, i.e., goal-directed action. As mentioned by most informants, the work on a specific task is done individually, and not together with others. Only at certain points during taskwork are others included, e.g., for code reviews, task-related discussions or when something impedes the taskwork. While doing taskwork, *feedback* can surface, either from the taskwork itself (e.g., real-time information from the tools or machines about the systems or what other team members are doing) or from other sources (e.g., from a manager or team member). There were also reported to be regular meetings that can drive team members to address evaluation and planning activities in the transition phase, taking the attention away from taskwork temporarily. The case analysis showed that *feedback* can turn into a *trigger*, something that prompts *adjustment* to taskwork or a more elaborate *evaluation and adaptation* of the task or planning activities.

*Trigger* represents the adaptation cue(s) that one or more team members recognize during taskwork. The findings support a general distinction between whether the trigger is perceived to have a high or low severity for the taskwork, where the main distinction is whether it moves the main attention towards transition processes or not. When the perceived solution to the trigger is under the SDTs control, it has low severity and relatively little adaptation is necessary. If it is outside the SDTs control, e.g., failures of other systems in the organization that the SDTs are dependent upon but do not control, the severity is high. Furthermore, having
other team members and relevant managers in close proximity, is generally perceived as beneficial both for continuous feedback, and for the response to triggers. This benefit is hampered if the team members who experience a trigger do not feel close to other members of (or belonging to) the product unit.

When the severity is perceived as low, adjustment to taskwork is made. Among else, adjustment involves seeking help, short task-related discussions with team members and coordinating work, which do not take the main attention away from goal-directed action. The case analysis suggests that the severity is perceived as low when the resources and expertise necessary to adapt and continue taskwork are readily available (either physically or virtually).

When the severity of a trigger is perceived as high, it prevents the continuation of taskwork and moves the main attention towards transition processes, i.e., the situation is evaluated and adaptation is made to either the task (e.g., change task) or planning (e.g., meeting activities and/or involving management). For example, this can happen when a higher priority task emerges, when the tools used for the work do not function (e.g., the software test/integration environments), or when dependent systems outside of the SDTs control are down or change. For the latter, the stage in the development process can determine whether it is a high or low severity trigger. The case analysis revealed that some tasks, coding wise, can be worked on and finished while dependent systems are out of function but when the code is to be tested and integrated to the product, then the dependent systems must work.

*Evaluate & adapt* refers to an assessment of the situation and how to proceed. It can involve multiple team members or be done individually. The case analysis revealed that if the current task is blocked, it can involve switching tasks for team members, create further tasks that needs to be addressed to solve the issue and/or going into planning activities like meetings and/or involve management. It was reported to also be routine meetings for the SDTs (e.g., weekly status and coordination meetings) which function like a forced visit to the transition phase, temporarily taking the main attention of software developers away from taskwork. In terms of adaptation, these meetings can both reveal matters that need to be dealt with but also be a way for team members to get a common idea of what others are working on and their progress that can be used to align their work.
Discussion

Team adaptation processes are usually presented as “cycles” (Baard et al., 2014), which we found to be the most suitable way of presenting the results of the case analysis as well. One key difference between some other team adaptation models (e.g., Burke et al., 2006; Maynard et al., 2015) and the suggested model (figure 3) is that we explicitly seek to include both the individual and team level, because that is what the findings suggests is most appropriate. That is, solely focusing on either one would in this study fail to capture the variation in the findings that explains how the SDTs adapt to task-based triggers. Another important difference is that the team adaptation model of Burke et al. (2006) presents the adaptation process in four phases, without any sensitivity to the severity of the triggers and subsequent adaptation processes. Thus, the current study brings the theory further by outlining potential differences in adaptation processes based on the trigger. The response to a high-severity trigger according to figure 3 is more similar to the generic response pattern of Burke et al. (2006). The results of the case analysis suggest that high-severity triggers lead to extensive evaluation and planning activities before attention is moved to taskwork again.

The challenges related to the complexity apparent in the findings, are some of the reasons why well-planned and formalized activities as means to achieve the desired goals in software development have been reported to be insufficient (Nerur & Balijepally, 2007). In this study, interdependencies with other systems that were outside of the immediate control of the SDTs were particularly challenging, for example because of the potential impact on the product of the SDTs by changes or failures in these systems. Such interdependencies with external entities and systems imply that the agility of SDTs can be severely constrained by what is external, if judged by the value added to the product of the SDTs. That points towards that multi-level research on agility in ISD is needed for a better understanding of the phenomenon. This is similar to what has been proposed in research on team effectiveness in general (Mathieu et al., 2008) and the team adaptation literature in particular (Burke et al., 2006). Thus, this study contributes by taking the different levels of analysis into account and by outlining potential ways in which the different levels interact.

Interdependencies between team members is commonly a core assumption underlying both teams and the need for teamwork (Kozlowski & Ilgen, 2006). The findings revealed that the SDTs seek to define tasks so that they can be worked on
individually from start to finish, with as little interdependencies between team members and external systems as possible. This implies that they seek to avoid unnecessary teamwork, which also suggests that individual capabilities should be especially important for performance (see Cockburn & Highsmith, 2001). Marks et al. (2001) emphasised that an effective team has both competent members and the appropriate team processes at any point in time. In this study, the planning for taskwork and adaptation triggers that occurred during taskwork often deemed teamwork necessary. The findings suggest that such task-based triggers happen frequently, which are the points in time where the interdependencies between team members and the systems become most apparent.

The findings showed that it was frequently the tasks and the involved activities that influenced who worked together as a team (the team boundaries) and the number of people involved in teamwork over time. This led to the interpretation underlying figure 3, that the organizing nature of teamwork revolves around the tasks. Given the continuous change perspective – team microdynamics – and emphasis on tasks, it seemed arbitrary to draw clear distinctions between team adaptation processes and team processes (teamwork). This is because the findings revealed that it was a task-based trigger that commonly led to the need for teamwork, since we considered low-severity triggers. Previous research seems to largely disregard the low-severity triggers and mostly focuses on the more novel, non-routine or disruptive triggers (e.g., Burke et al., 2006; Kude et al., 2014). We included what others (e.g., Kude et al., 2014) have treated as “routine” triggers that can be solved by a routine response. This can contribute to a more holistic view of the team adaptation process and thereby bring the theory one step further.

The findings supported the idea that some triggers have higher or lower severity than others, and by including that aspect, the adaptation processes that occur regularly for the SDTs are addressed. This ties in with how Christian et al. (2017) differentiate between routine team performance (completing similar tasks over time) and adaptive team performance (new tasks requiring new actions). The case analysis revealed that making such a distinction would not make sense, given the aspect of the severity of the triggers. This might also be tied to the case context, since “problem solving” was considered the way of working, rather than following plans, because of the frequency of changes and difficulty of planning up-front. This implies that the feedback that occurs during taskwork (which may become a trigger) is necessary for revealing task-relevant information that brings the problem-solving
efforts further. As apparent in the findings, tasks that at first seem routine may after engaging in taskwork turn out to contain novel, or “non-routine” aspects which requires adaptation.

Kude et al. (2014) identified three categories of potential “non-routine” triggers, i.e., technical volatilility (e.g., requirement changes), technological disruption (e.g., issues with internal and/or external technological systems) and team instability (stability of team membership). The findings in the current research support the aspect related to technical volatility, in the form of the business side changing priorities, and technological disruption in the form of issues with the technological systems the SDTs were dependent upon. Additionally, this research showed that a crucial aspect to the severity of the latter was whether the SDTs had control over the systems themselves and/or the potential solution. The stability of team membership can affect how the SDTs adapt (Kude et al., 2014) and is similar to what Maynard et al. (2015) more broadly label team-based triggers, which entails that it is outside of the scope of this thesis to address. Based on the adaptation patterns Kude et al. (2014) found, the current study supports their general notion that the adaptation triggers which have a large impact lead to more extensive adaptation processes. We were further able to distinguish between two responses in terms of team processes depending on the severity of the trigger.

According to the findings, we were able to distinguish two phenomena which have been referred to as emergent states in the literature (Marks et al., 2001). These describe “cognitive, motivational, and affective states of teams, as opposed to the nature of their member interaction” (Marks et al., 2001, p. 357). The first one resembles what has been called shared mental models in the literature (see Mathieu, Heffner, Goodwin, Salas, & Cannon-Bowers, 2000). Shared mental models have been tied to improving collaboration during the software development process in agile software development (Yu & Petter, 2014). The findings revealed that knowing about each other’s expertise supported the developers in cooperating and therefore in handling interdependencies. In contrast, not knowing about the skills of the developers in other squads was reported to impede handling interdependencies amongst them. When looking at planning, it was stated by the informants that having all involved parties aligned supported the process of planning. This can be seen as similar to the understanding of shared mental models of Stout, Cannon-Bowers, Salas, and Milanovich (1999, p. 61), who refer to them as cognitive representations that are “thought to provide the team members with a
common understanding of who is responsible for what task and what the information requirements are”. Rosen et al. (2011) mention that shared mental models contribute to a shared understanding of situations, which are needed to adapt. Even though the findings suggest that such a shared understanding can be helpful, future research is needed in order to further understand this emergent state in the adaptation processes of SDTs.

The second emergent state which can be interpreted from the findings resembles psychological safety, which has been identified as playing an important part in team adaptation by for example Rosen et al. (2011). They describe this phenomenon as an emergent state denoting the shared belief that it is safe for interpersonal risk-taking in the team. The findings of the current study revealed that feelings of social connectedness that lowered the threshold of approaching others was important during planning activities, as well as during taskwork. This is similar to Rosen et al. (2011), who state it is especially important during what they label “plan formulation” and “plan execution”. Several informants reported that the benefits of being located on the same floor as the rest of the unit (which is supposed to facilitate team member interaction) did not come to life before a certain social connectedness was established. In this sense, psychological safety can be seen as a necessary prerequisite of the positive effects of co-location in this study. Future research is needed to gain a better understanding of such a relationship.
Implications, Limitations and Future Research

The implications of this thesis will be presented by distinguishing between implications that are practical (i.e., important for managers and other practitioners) and theoretical (i.e., relevant for academia). Possible directions for future research will be suggested in combination with both the theoretical implications and the limitations of this study. It should be mentioned that we do not claim any of the findings to be generalizable to other contexts, given the research design.

Practical and Theoretical Implications

This thesis offers several practical implications. Firstly, we have outlined certain team processes which are part of how the agile SDTs in this case adapt to task-based triggers. These team processes have been proven by earlier research to be important for team performance outcomes (e.g., LePine et al., 2008). Figure 3 conceptualises these team processes as derived from the findings. The model’s focus on tasks and its continuous change perspective bring it closer to the daily activities of SDTs. It can help identifying crucial aspects of the team adaptation processes which can support practitioners in their efforts of improving their team’s adaptive performance.

Secondly, this study indicates that the SDTs can be constrained by external systems and/or departments that are not under their control. This means that practitioners should take into account externalities when considering how to improve the adaptive performance of their teams.

A third implication for practitioners is that psychological safety can play an important role when co-location is supposed to increase interactions between team members. The findings show that co-located team members had a higher threshold for approaching others when they needed help if they did not have a feeling of social connection to them. This underlines the importance of social events in order to facilitate the development of a social connection.

This thesis also contains multiple theoretical implications. Firstly, this study showed that considering the dynamics between the individual and team level in studies of team adaptation simultaneously can be beneficial. Within the broader team adaptation literature, many consider it necessary to employ a multi-level and temporal perspective to understand the phenomena better, i.e., similar to what we have called a micro-dynamic approach earlier. For example, Burke et al. (2006) state that adaptive performance is explained by the patterns of individual team
member action towards goal-accomplishment. Similarly, Rosen et al. (2011) mention that one should be sensitive to shifts in individual performance, not only on the aggregate team level. Baard et al. (2014) reviewed both literature on individual- and team level adaptation and proposed that an integration was necessary.

Secondly, this study investigated the type of trigger and its severity in a specific context, as proposed by Maynard et al. (2015). We have added some preliminary evidence for the proposition of Maynard et al. (2015), that low-severity task-based triggers prompt teams to address their action processes, while high-severity triggers prompt teams to first attend transition processes, before continuing on action processes.

Thirdly, the finding of this study with regards to that the teams organize dynamically based on the task, i.e., the de-emphasising of clear team boundaries, should be of interest for future research within the domain of ISD. Similar to the arguments of Humphrey and Aime (2014), we challenge the assumption that teams should be treated as an entity with stable boundaries. If these boundaries are not considered as dynamic, one may miss the nature of interaction between the SDTs and the individuals within. This can be particularly important for future quantitative work on teams, since one should consider whether a team in a certain context can be treated as a single measurable unit with relatively stable boundaries or not.

A fourth theoretical implication regards the finding that the SDTs seek to define tasks such that they may be solved individually. This finding implies that future research could benefit from investigating individual adaptability (see for example Baard et al., 2014) or other relevant individual characteristics within the domain of ISD.

The fifth theoretical implication is that this study suggests that utilising theory from the literature on team adaptation can be fruitful for advancing the theory on agility in ISD. Thus, we answer the call from researchers with regards to the lack of theoretical foundation of agility in ISD (see for example Conboy, 2009; Werder & Maedche, 2018).

**Limitations and Future Research**

The first limitation of this study is that even though we were able to gather process data by making the informants describe their perceptions in the form of stories, the scope of this thesis prevented us from gathering longitudinal data.
Similarly to what Baard et al. (2014) and Werder and Maedche (2018) propose, we suggest that future research should make use of longitudinal designs in order to investigate team adaptation from a continuous change perspective. Doing so in other organizations and contexts would contribute to that the findings in this study could be generalized.

A second limitation is that this research mainly focused on adaptation and change with a focus on negative triggers, e.g., as "challenges", "issues", or "problems". A reason for this might be the way we phrased the questions in the interviews, even though we tried to do that in a neutral way. Another reason could be the general association of people towards “change”, which is often reported to be negative. Regardless of the reason for this study’s focus on negative triggers, future research can leverage on an inquiry into the opportunities in relation to adaptation and change, by actively focusing on positive triggers.

A third limitation regards the academic background of the researchers. In line with what Lenberg et al. (2015) propose, future research can benefit of including researchers with both an ISD-background and a background from social sciences. This could add to the credibility and the understanding of the nuances in the data, which could reveal further intricacies.
Conclusion

This study investigated how agile software development teams in a Nordic financial institution adapt to task-based triggers. The studied agile software development teams seek to define tasks in a manner that they can be worked on individually. We found that it is mainly the adaptation triggers that occur during taskwork which prompts task-related interaction. The findings showed that these adaptation triggers can either prompt small adjustments to taskwork or be more severe and move the focus to evaluation and planning activities before taskwork is continued. The more severe triggers are typically changes or failures in interdependent systems and departments in the organization, as well as changes in priorities from the management. Based on the findings, we proposed a model (figure 3) describing how the agile software development teams adapt to task-based triggers. This cycle-model suggests that the task is pivotal in terms of how the teams adapt and describes a process of six stages and the relationship between them. Future research is needed in order to prove the potential generalisability of the findings and the proposed model.
References


http://aisel.aisnet.org/ecis2014/proceedings/track13/11


Appendix

Appendix A: Interview guide

Introduction

*Note:* We will introduce our research project and inform the interviewees about the processing of data, anonymisation and the use of quotes. Make clear how the participant contributes to the research project.

- “Which team are you working in and what is your current role?”
- “How long have you had this role?”
- “Have you worked in this role or a similar role before? If yes, how long?”

How does the team work?

*Note:* get an impression how the team looks like and how it works. Ask questions related to these notes, if relevant.

- What’s the release-to-customer in your team?
- Team size and team member stability
- Task complexity and task interdependence
- Project design, product and development process
- Organization of tasks

Trigger / team processes / outcomes

*Note:* Facilitate a conversation about the topics and uncover the aspects that the informant perceives as most important. Elicit extended storytelling about the topics.

- “What are the main sources of interruptions while you and your team are working?” (e.g., which can lead to changes in team processes or technology)
  - “Can you describe an example of such an event which happened to your team in the past?”
    - “How did you and/or your team react to / work through that event?”
    - “How did the way your team reacted to / worked through that event affect your team?” (e.g. in team effectiveness, performance, interpersonal relationships)
      - “Why do you think that is?” / “What would you think would have been better to do?”

Ending

Summarise the main statements of the interview (also do this continuously in order to validate meaning)

Repeat the context of use and the formalities

Thank the informant for participating