Inter-team Coordination in Large-Scale Agile Software Development Projects

Tomas Gustavsson

Faculty of Arts and Social Sciences

Information Systems

DOCTORAL THESIS | Karlstad University Studies | 2020:30

Inter-team Coordination in Large-Scale Agile Software Development Projects

Tomas Gustavsson

DOCTORAL THESIS | Karlstad University Studies | 2020:30

Inter-team Coordination in Large-Scale Agile Software Development Projects

Tomas Gustavsson

DOCTORAL THESIS

Karlstad University Studies | 2020:30

urn:nbn:se:kau:diva-80195

ISSN 1403-8099

ISBN 978-91-7867-151-9 (print)

ISBN 978-91-7867-155-7 (pdf)

Distribution:
Karlstad University
Faculty of Arts and Social Sciences
Karlstad Business School
SE-651 88 Karlstad, Sweden
+46 54 700 10 00

Print: Universitetstryckeriet, Karlstad 2020

WWW.KAU.SE

Abstract

Software development organizations worldwide are adopting values, principles, and frameworks to implement Agile ways of working. The advantages of Agile ways of working are seen in teams that are allowed a high level of autonomy. The Agile methods were initially designed for use in small, single-team projects, and routines for coordination between several teams have not been adopted in the same way as routines for coordination within the team. With several teams coordinating, autonomy must, to some extent, be sacrificed in the individual team. Work needs to be coordinated with other teams, and a project is often part of a portfolio or program. The purpose of this research is to investigate routines for inter-team coordination: how they are performed; if, how and why they are tailored, and the impacts of these added routines in relation to Agile values and principles, in particular team autonomy.

This thesis is based on empirical studies at three organizations with disparate business logics. One is a product development department in the automotive industry, one is a business bank, and one is an IT department at a Swedish government agency. Data has been collected from 379 hours of on-site observations, 28 interviews, and 201 answers to a survey questionnaire.

Insights from these cases build on coordination theories as well as institutional logics (new institutional theory). One contribution of this thesis is the rich descriptions of tailoring of inter-team coordination routines. Another contribution is the identified perceived impacts of the implemented inter-team coordination routines, especially regarding perceived changes to team autonomy. An important theoretical contribution is the identified and defined institutional logics (*Agile toolbox logic, Agile rulebook logic, Flow efficiency logic,* and *Resource efficiency logic*), which can be used for analysis of large-scale Agile software development projects.

Keywords: Inter-Team Coordination, Large-Scale Agile Software Development, Institutional logics, Scaled Agile Framework, Project Management.

Contents

Preface

In 1996, after three wonderful years of being a student at Karlstad University, I started working as a software developer in different roles in various places such as Gothenburg, Copenhagen, and Stockholm. I returned to Karlstad in 2003 and was employed as a part-time lecturer at Karlstad University, part-time working with my own business as a consultant to help organizations mostly within IT project management. Sometimes I helped organizations to implement 'the new thing' called Agile.

Since then, Agile ways of working have become a commodity in the software industry and is growing increasingly popular even in other business areas. Being a part of the Agile movement in Sweden from the very beginning, I am both blessed and possibly somewhat biased. I am blessed in the sense that I have found a fascinating area that I love to teach about and to learn more about. And there is still so much to learn.

Since I had the opportunity to enroll as a Ph.D. student in 2016, I have (as most Ph.D. students, I have been told) been torn between knowing exactly what the aim of my research is and not having a clue about whether I am on the right track or not. Students tackle this problem in different ways. Some are guided by a specific theory that they want to develop but have a hard time understanding how to explore and develop it; others want to study a phenomenon but have a hard time finding a proper theory to help them understand and develop knowledge about it.

For me, the phenomenon itself that has been at the center of my thoughts. As Agile ways of working, intentionally developed for small projects with few team members, have become increasingly popular even in larger organizations, coordination between several teams in these larger settings has so far not been much explored. Many of the practices and routines described in Agile methods and frameworks are helpful to gain the benefits from becoming an autonomous team. However, when autonomous teams need to work together, they must sacrifice some level of autonomy since work needs to be coordinated with other teams. So how could this be balanced? How could we get the best out of each team and still coordinate work to benefit the whole project or organization? What happens to the team members when the level of autonomy changes? This phenomenon, *how inter-team coordination is performed and its implications in a large-scale Agile work setting,* has been my guiding light throughout the years of writing my thesis.

Knowing what to explore, based on a phenomenon instead of a specific theory, has naturally turned my work into an abductive methodological quest. Since Agile software development is a rather young research field and large-scale Agile software development is practically an infant, there have not been many studies to lean on in finding "natural" theories or methods to use.

From being a practitioner and, as previously mentioned, possibly being biased regarding the benefits of Agile software development, I now have put on my academic glasses to turn a critical eye on the Agile movement. My background gives me an advantage since I already understand a lot about the field, the work settings, the philosophy, and the methods that I have observed. I have investigated both drawbacks and benefits, searched for theories to explain what is happening, and developed contributions to improve our theoretical understanding.

Acknowledgements

In finalizing my thesis, there are so many people who have helped me with inspiration, motivation, innovation, and, of course, perspiration. I would like to express my deepest gratitude to all of you.

I would like to start by acknowledging what a privilege it is to be surrounded by such wonderful colleagues. The Project Management lecturers at Karlstad University: Lennart, Carin, Henrik, Lasse, Johan, and recently Erik, Tove, and John - you are my Dream Team! Tomas Jansson, Peter Rönnlund, and Per Strömgren – I miss you, my former comrades in arms.

I could not have done this without my supporting, constructive and critical supervisors. Thank you for helping me along the journey Linda Bergkvist, Marie-Therese Christiansson, and John Sören Pettersson. You have not only guided me but also trusted me with a considerable portion of freedom in my research. Feedback is always important. Besides my supervisors, I would like to thank Lucas Gren for being discussant in my first seminar and Ovais Ahmad for being discussant in my final seminar. I would like to thank all the people who provided feedback at the conferences with the Swedish Research School of Management and Information Technology (MIT). The biannual conferences offered great insights.

There are so many that have improved my research competence during these years. Thank you so much for taking your time in helping me, Raul Ferrer Conill (KaU), Jari Appelgren (KaU), Niklas Jakobsson (KaU), Claes Thorén (UU), Fredrik Karlsson (ÖrU), Kai Wistrand (ÖrU), and all the colleagues at Informatik at Karlstad University. I would also like to express my sincerest thanks to the case organizations who invited me to observe and disturb in their daily activities.

Writing a doctoral thesis is sometimes a lonely task. Therefore, I am so glad that I, once a month, could attend the Writing Boot Camp for a day of joint struggle against writer's block. So thank you, recruits: Ulrik, Åsa, Anna H, Anna W, Helena, Håkan, Kristoffer, Charlotte and Martina.

Finally, I am grateful for the love and support of my family and friends. My parents, Rune and Gun, thank you for encouraging me to do what I want. Maria, my love, I am so grateful that you are in my life! Frida and Klara, you are the best bonus-daughters one could have ever asked for.

Karlstad, September 2020

Tomas Gustavsson

List of figures

List of tables

Recurring abbreviations

AP	Agile Principle
ASD	Agile Software Development
IS	Information Systems
ISD	Information Systems Development
IT	Information Technology
LeSS	Large-Scale Scrum
PI	Program Increment
PO	Product Owner
RTE	Release Train Engineer
SAFe	Scaled Agile Framework (© Scaled Agile, Inc.)
SLR	Systematic Literature Review
SM	Scrum Master
SoS	Scrum of Scrums
XP	eXtreme Programming

1 Introduction

Software development projects are often complex undertakings that involve multiple interdependencies between resources, tasks, teams, roles, and various software components and systems (Bathallath, Smedberg, & Kjellin, 2016; Gustavsson & Görling, 2019). Coordination of interdependencies is one of the biggest challenges associated with large-scale software development today (Šāblis, Šmite, & Moe, 2020).

Coordination was early identified as a particular challenge in software development projects. In the 1990s, software projects were often associated with overruns on time and cost, and many referred to the *software crisis* (Kraut & Streeter, 1995). Kraut and Streeter (1995, p. 69) stated: "While there is no single cause of the software crisis, a major contribution is the problem of coordinating activities while developing large software systems".

IT project management was during that time seen as a subset of general management with a highly technocratic and rationalistic perspective, focusing on control, heavyweight documentation and comprehensive planning (Packendorff, 1995). Software developers often reacted negatively to the heavyweight document and plan-driven approaches. The rapidly changing environment made it difficult to initially gain an overview of the development process (Karrbom Gustavsson & Hallin, 2014).

After the software crisis, new methods for software development were developed and disseminated in the IT industry. New *lightweight* software development methods and frameworks emerged (Cockburn & Highsmith, 2001; Dingsøyr, Nerur, Balijepally, & Moe, 2012; Gustavsson, 2007), such as eXtreme Programming (XP) and Scrum. These were lightweight in the sense that much of the processes, practices, and routines where not described in detail. Hence, the documentation of the frameworks is thin, e.g., the Scrum Guide (Sutherland & Schwaber, 2013) which contains only 19 pages. With only a few pages describing the methods, much of the details were left to the implementing organizations to decide on. These methods were later renamed to *Agile* methods. Agile is spelled with a capital "A" since it is the name of the methods, not to be mistaken with *agile*, which is an embedded trait or attribute characterized by durability, resilience, speed, flexibility, attunement and preparedness (Prosci, 2020).

The origin of the term *Agile* dates back to 2001. The name was chosen by seventeen of the inventors of new methods who met and discussed the current state of software development (Beck et al., 2001). The essence of

these methods was the ability to embrace change, but also continuous learning, coordination, and communication within the team (Šmite, Moe, & Ågerfalk, 2010). Since then, Agile Software Development (ASD) has become the norm within software development (Dingsøyr et al., 2012; Licorish et al., 2016; VersionOne, 2020). This evolution is not very surprising since some of the benefits of ASD proved by scientific researchers are an increase in quality, productivity and enhanced flexibility (Campanelli & Parreiras, 2015).

Practices in the IT industry have also inspired the overall project management discipline (Conforto et al., 2014). The change entails seeing people as the primary drivers and a stronger focus on *soft* factors (Karrbom Gustavsson & Hallin, 2014). However, allowing power to self-organizing teams was at first somewhat of a culture shock to project managers (Karrbom Gustavsson & Hallin, 2014; Karlström & Runeson, 2005).

Agile ways of working have constantly been disseminated to new industries and businesses (Gustavsson & Rönnlund, 2010; Gustavsson, 2016). Therefore, ASD is not only common within private companies but in public organizations as well. According to a recent study of Swedish government agencies, 87.8% (65 out of 73 responding agencies) reported that their software development is more Agile than plan-driven (Borg, Olsson, Franke, & Assar, 2018).

The Agile methods were originally intended for small, self-managing, and co-located teams. Scrum (Schwaber & Beedle, 2002), the most commonly adopted Agile method (VersionOne, 2020), was based on studies on high-performance teams by Takeuchi and Nonaka (1986). Their studies suggest that projects using small, cross-functional teams produce the best results (Hoda, Noble, & Marshall, 2012; Moe, Stray, & Hoda, 2019; Sutherland & Schwaber, 2011). That is why ASD advocates self-organizing teams that are allowed high levels of autonomy (Hoda & Murugesan, 2016). ASD team members are known to take on informal and spontaneous roles to satisfy various organizing needs of the team (Hoda & Murugesan, 2016) and perform balancing acts between freedom and responsibility (Hoda et al., 2012). This also radically changes the role of the project manager, who must place more trust in the team to make the right decisions (McHugh, Conboy, & Lang, 2011). Thereby, project management becomes a question not only for the project manager but instead a joint effort together with the team (McHugh et al., 2011).

The popularity of these methods has spurred their use also in large development projects (Xu, 2009; Hobbs & Petit, 2017; Klünder, Hohl &

Schneider, 2018) but Scrum does not cover large-scale organizational aspects (Maples, 2009), only how to work in single teams. With several teams cooperating, there is not only a need for coordination within the team but between teams, which is called *inter-team coordination* (Moe, Holmström Olsson, & Dingsøyr, 2016). To clarify, here is a typical example of a dependency that needs inter-team coordination: Team A is supposed to develop functionality X. But in order to do that, a functionality called Y needs to be finished. The necessary coordination in this example means that another team needs to be responsible for developing functionality Y (which team?) and Team A needs to be informed about when the expected functionality will be finished. Also, Team A needs to be informed if the expected functionality is delayed.

The traditional project management approach is to focus on formal coordinating mechanisms through a pre-defined process, precise planning, and high levels of specialization in role assignments (Nerur, Mahapatra, & Mangalaraj, 2005). Typically, the project management role is responsible for coordination between teams. In ASD, the responsibility for coordination is not appointed to a single role, but instead expected to be dealt with by teams and different roles depending on the level of coordination needed (McHugh, Conboy, & Lang, 2011; Gustavsson, 2017).

Processes, practices, and routines which worked well for a small team could be difficult to scale to an entire organization (Maples, 2009). One reason for this difficulty could be the view on processes within the organization. Cockburn and Highsmith (2001) claim that processes are often seen as a way to standardize how people work in the organization, while the Agile approach is to capitalize on each individual and each team's unique strengths. This is why, according to Cockburn and Highsmith (2001), every Agile process, practice, and routine must be selected, tailored, and adapted in every single team.

According to The State of Agile survey (VersionOne, 2020), the IT industry trend today is to implement large-scale ASD frameworks, even in larger organizations. Although the survey is nonscientific and problematic from a methodological point of view (Stavru, 2014), it is the largest reoccurring survey on agile adoption. Despite the claimed proper approach of *tailoring everything* suggested by Cockburn and Highsmith (2001), large-scale frameworks are often prescriptive in extensive detail. Routines for inter-team coordination are presented with standard agendas, time limits for meetings, and which roles should attend.

It is, of course, up to the organization to decide how much the coordination routines of the frameworks should be tailored. However, one common criticism of the frameworks, both by practitioners and academics, is the detailed prescriptions of how to work in a large-scale Agile setting (Schwaber, 2013; Alqudah & Razali, 2016; Stojanov, Turetken, & Trienekens, 2015). Critics claim there are risks in following these prescriptions instead of tailoring to your own needs. The understanding of tailoring in large-scale ASD projects is limited, as well as the impacts of tailoring or not tailoring (Dingsøyr et al., 2018). Although tailoring software development methods have always been at the core of the Information Systems (IS) research field (Ågerfalk, Fitzgerald, & Slaughter, 2009), more research on implementing and tailoring routines for inter-team coordination in large-scale projects is called for (Dietrich, Kujala, & Artto, 2013; Dingsøyr et al., 2018).

Practices and routines for coordination has been an important research topic in organizational and management studies for a long time (March & Simon, 1958; Malone & Crowston, 1994; Mintzberg, 1983; Van de Ven, Delbecq, & Koenig, 1976). Critique on this earlier work (Jarzabkowski, Lê, & Feldman, 2012; Okhuysen & Bechky, 2009) is that there has been too much of a static view of coordination. How to solve emerging dependencies has been mostly overlooked, and routines to manage them are rarely explained. One reason for this could be that the routines have been difficult for researchers to measure and have therefore remained largely unexamined. This is unfortunate, since emerging dependency issues and the responses to them are important for efficiency in organizations (Okhuysen, & Bechky, 2009). The lack of studies within this area is also prevalent in IS research. Therefore, more research on coordination to manage emerging dependency issues is called for in IS research as well (Espinosa, Slaughter, Kraut, & Herbsleb, 2007; Taxén, & Riedl, 2016).

1.1 Why study inter-team coordination in ASD?

The Agile Manifesto (Beck et al., 2001) stresses the importance of self-organizing teams, allowing autonomy and trust in the team's ability to make wise decisions, solve problems, and deliver results. Allowing autonomy to the team is a major reason for success in Agile development (Cockburn & Highsmith, 2001; Jansson, 2015; Stray, Moe, & Hoda, 2018). As discussed above, Scrum was originally inspired by Takeuchi and

Nonaka's (1986) article in Harvard Business Review. The researchers investigated product development in Japanese corporations where *self-organizing project teams* was one of six key characteristics for success. Research in other industries also confirms that autonomous and empowered teams are more productive, more proactive, have higher levels of team satisfaction, and team commitment (Kirkman & Rosen, 1999).

In a large-scale ASD setting, where several teams cooperate towards a common goal, new problems arise, such as dependencies between activities and aligning of goals between teams. Typically, a software development team needs to coordinate with other teams when it comes to constraints of, for example, requirements, testing, and integration (Šāblis et al., 2020). These dependencies limit the amount of autonomy and empowerment to the individual team. The need to align the work process with the rest of the organization reduces team autonomy (Gundelsby, 2018). As Bass and Haxby (2019) explain:

> "As soon as self-organizing teams work together, they must sacrifice some level of autonomy. Feature delivery needs to be coordinated with other teams, and a project is often part of a portfolio of related development projects" (p. 58).

Thus, the balance between the benefits of autonomous, empowered teams versus alignment towards a common goal is an important problem to solve for the software industry today (Moe et al., 2016).

Although research into the Agile approach to software development has matured in recent years, several open questions remain. The relevance and implications of certain fundamental organizational concepts are still not fully understood (Ågerfalk et al., 2009). One such concept is coordination. ASD in large-scale settings in general (Dingsøyr et al., 2012) and inter-team coordination in particular (Moe et al., 2016) are seen as important areas for further research.

Hoegl, Weinkauf, and Gemuenden (2004) point out that coordination is more important to team performance in large projects, with several cooperating teams than in one-team projects. In the case of high uncertainty in multi-team projects, the work relies heavily on coordination through group mode (Dietrich et al., 2013; Scheerer, Hildenbrand, & Kude, 2014), i.e., the coordination between teams, also referred to as *inter-team coordination.* To maintain the balance between alignment and autonomy in the teams, routines for inter-team coordination have been proposed in the Agile community to reduce negative impacts while maintaining the posi-

tive impacts of teamwork from the Agile ways of working (Moe et al., 2016). But do these inter-team coordination routines really have the intended impacts on the teams and their team members? Deciding on what coordination routines to use and how to use them is important, as they have a significant influence on information sharing, workflow fluency between teams, the efficiency of the project, and learning outcomes (Dietrich et al., 2013).

According to an annually recurring industry survey (VersionOne, 2020), the most commonly adopted framework for large-scale Agile ways of working today is, by far, the Scaled Agile Framework (SAFe). This framework prescribes a number of routines for inter-team coordination. The framework originator, Leffingwell et al. (2017), makes several claims regarding expected beneficial impacts by implementing SAFe, such as employee engagement, productivity increases of 20 to 50 percent, and even an increase of time to market with 30 to 75 percent, while no drawbacks of implementing SAFe are mentioned. SAFe has been criticized by several Agile practitioners (Maximini, 2013; Schwaber, 2013) and researchers (Alqudah & Razali, 2016; Stojanov et al., 2015) due to the risk of delimiting autonomy to the individual team. Limiting team autonomy could have negative impacts on teamwork and the performance of the team. However, these supposed negative impacts when implementing inter-team coordination routines have not been investigated in large-scale ASD (Uludağ et al., 2020).

1.2 Purpose and research questions

Agile ways of working in large-scale settings have not been much studied since it is such a new phenomenon, hence the knowledge gap in this area (Dingsøyr, 2012, Moe et al., 2016). Further research on impacts of the use of Agile methods is needed in the industry (Laanti, 2012). We know very little about how Agile values and principles are affected by scaling up and how team members working at the core of it are affected (Moe et al., 2016). What we do know is that many organizations working with software development are jumping on the bandwagon to adopt large-scale Agile frameworks (Laanti & Kettunen, 2019; Moe et al., 2016; VersionOne, 2020), and several of them are government agencies using taxpayers' money to accomplish this shift. To address the knowledge gap, the overall purpose of the research presented in this thesis is to describe and explain how inter-team coordination is conducted in large-scale ASD pro-

jects. The purpose is also to investigate why the chosen routines were used and tailored, and what the impacts on Agile values and principles are. In particular, since it is important to sustain a high level of team autonomy (Moe, Dingsøyr, & Dybå, 2009), the impact on team autonomy is of much interest.

To pursue this purpose, the following research questions are answered in this thesis:

RQ1: How are inter-team coordination routines performed and tailored in large-scale Agile software development?

RQ2: Why are inter-team coordination routines tailored (or not tailored)?

RQ3: What perceived impacts (benefits and drawbacks) are associated with the implementation of inter-team coordination routines?

Since the impact on team autonomy is of particular interest to investigate, the fourth research question reads:

RQ4: How is team autonomy perceived to be changed after implementing inter-team coordination routines?

The presented purpose and research questions express that this study follows a *practice research approach*. Goldkuhl (2012, p. 65) argues that "in IS and other related disciplines we should study *practices*. This means that we should acknowledge the practice character of the empirical field". A practice is defined by Goldkuhl (2012) as a "meaningful and coherent assemblage of human actors, actions, utterances and documents, and material artefacts" (p. 66). From this research point of view, practice relates to the routines performed between teams for the purpose of coordinating software development work. This is sometimes called 'the practice turn' in science (Goldkuhl, 2012).

The creation of abstract and useful knowledge to *improve existence through knowledge* is the key motivator for the practice research approach, but also to contribute to the research community as additions to the scientific body of knowledge (Goldkuhl, 2012). In project management research, this approach is named project-as-practice (Blomquist, Hällgren, Nilsson, & Söderholm, 2010). The focus is to investigate the micro-level practices and routines performed in projects, attempting to understand how practitioners act and make sense of their situation. The activities performed in projects are studied, as well as how these align with or deviate from established norms, routines, and behavioral expectations (Manning,

2008). The approach involves conceptualizing norms and routines as potentially dynamic, contextual, and subject to change.

1.3 Central concepts and scope to frame the study

The central concepts used in this thesis are visually presented in Figure 1, and some of them are explained in this chapter to clarify the scope of the study. This section also presents concepts that are *not* part of the study for this thesis, since several concepts are quite close and similar. The concepts that are not explained in this chapter will be described in further detail in the following chapters.

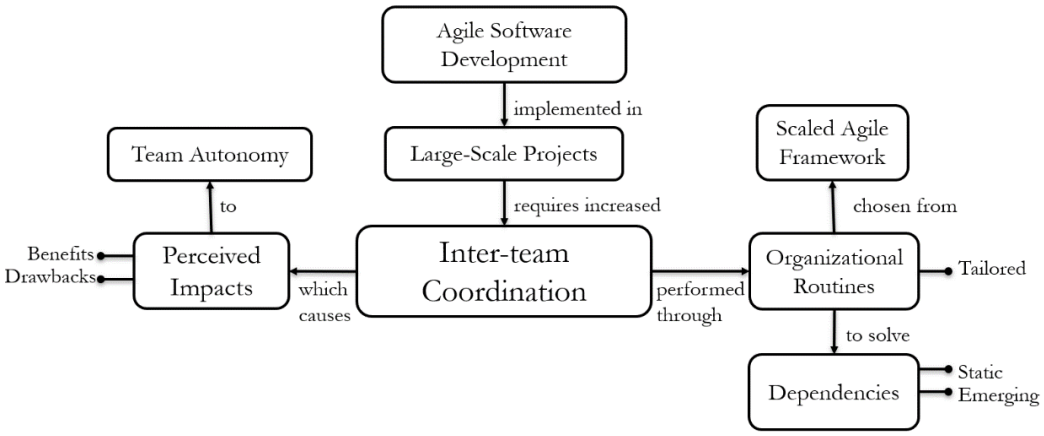

Figure 1. Central concepts.

First, a walkthrough of Figure 1 on how to read it: Starting from the top, the concept area to be investigated is *Agile Software Development*, implemented in *large-scale projects*. Software development methods belonging to the pre-dominant era, called 'Waterfall' or plan-driven methods (van Waardenburg & van Vliet, 2013), are not of interest for this study. Neither are small-scale projects, no matter which development methodology is chosen, because there will not be a need for inter-team coordination routines where the project consists of only one team. What defines large-scale in Agile software development, as opposed to small-scale, is further explained and discussed in chapter 2.4. Another important concept sometimes mentioned in discussions regarding how to scale Agile ways of working is *Lean*, the mindset and methods originating from the Toyota Production System. Although they are in many ways similar, the main focus of Lean is on *efficiency,* while the main focus in Agile is on *effectiveness*, to produce the desired output (Agile Alliance, 2001; Scaled Agile, 2020). Also, Lean originates from the manufacturing sector, not software devel-

opment. Therefore, references to Lean are kept to a minimum in this thesis. When there was a choice as to whether to refer to a Lean or Agile reference, the latter was selected.

Moving right in Figure 1, the investigated inter-team coordination is performed through *organizational routines*. These are sometimes referred to as practices, mechanisms, or even processes, but I use the definition of organizational routines (hereafter, simply *routines*) according to Feldman and Pentland (2003, p. 94): "repetitive, recognizable patterns of interdependent actions, carried out by multiple actors". To clarify, the focus is on several people performing the coordination. The word *practice* is sometimes used, but a practice could also be something performed by an individual on his or her own. Therefore, I use the term routine according to the above definition to specify that the focus of this study is actions between actors. This also means that I do not study automized, computer-based coordination, but instead, what *people* do to coordinate.

The organizational routines could also be more or less tailored, e.g., as opposed to the originally intended approach suggested by a framework such as the *Scaled Agile Framework* (SAFe). SAFe is further described in chapter 2.3. The routines could also be tailored along the way, as opposed to the original implementation. The sensemaking behind decisions made on tailoring could differ between organizations. To better understand *why* organizations make different decisions, one way is to investigate how decision makers adhere to different institutional logics (Thornton, Ocasio, & Lounsbury, 2012). Institutional logics are concepts defined in new institutional theory (Thornton & Ocasio, 2008) and is further explained in chapter 4.

These inter-team coordination routines are intended to solve *dependencies* between teams (Moe et al., 2018). Such dependencies could be *static* where coordination can be planned in advance, or *emerging*, which means that they appear unexpectedly (Okhuysen & Bechky, 2009). The importance of being able to identify both kinds and especially to solve emerging dependencies is highlighted by several researchers (Jarzabkowskij et al. 2012; Okhuysen & Bechky, 2009) and is further described in chapter 3.1.

Moving to the left in Figure 1, implementing new inter-team coordination routines in an organization causes impacts. This study investigates the *perceived impacts*, both benefits, and drawbacks, on the people working in large-scale ASD projects. The study does not measure actual impacts but investigates the perceived impacts by employees. One such impact of

specific interest is the perceived changes to team autonomy because highly autonomous teams are more productive, and proactive, and have higher levels of team satisfaction and team commitment (Takeuchi, & Nonaka, 1986; Kirkman & Rosen, 1999).

The longitudinal case studies conducted in this thesis have taken place in three organizations. Regarding the concept of *tailoring* the routines, the focus on tailoring in this study is limited to the starting point of implementing a large-scale Agile framework and during the first two years from the starting point of implementation in the case organizations. The Agile ways of working in the organizations have, of course, been developed further – and are still being developed. The findings in this study, however, come from data from these two first years of tailoring.

1.4 Thesis structure

After this introduction to the research topic, chapter two gives a presentation of Agile ways of working; the values, principles, and previous research. In the following chapters, I present and motivate the theoretical framework that guides the undertaken study. The third chapter presents coordination theory, and in the fourth chapter, I describe new institutional theory (and institutional logics in particular). In chapter five, the applied research methods used in the studies are described and motivated in detail. The sixth chapter presents four institutional logics which are used for analyses of the cases. The following three chapters present how inter-team coordination was performed in the three case organizations. Chapter ten presents a cross-case comparison and synthesis of the three studied cases. Finally, chapter eleven contains final discussions and conclusions.

2 Agile Software Development

The aim of this chapter is to describe the values, principles, practices, and routines in ASD. The reason for presenting this is twofold; one reason is to show how ASD differs from other ways of thinking and working. The other reason is to present previous research to explain what we know about ASD and the knowledge gaps that exist today. This section is divided into four subsections, where the first part briefly describes the background of software development and specifically highlights the rise of Agile methods and frameworks. The next section presents the most implemented framework, Scrum. In the following section, SAFe, the most implemented large-scale framework, is discussed. The next section presents definitions of large-scale ASD, and the final section presents previous research on large-scale ASD.

2.1 Software development and the rise of Agile

One might call the evolution of computers something of a paradox: when the capabilities of microprocessors have been growing exponentially (double the speed in eighteen months), software development speed has only grown linearly. The problem of developing software, with longer and more complex development projects, was identified in 1968 and named the Software Crisis (Kraut & Streeter, 1995). Several tools have been developed in order to solve the problem, such as 3rd-generation programming languages and object-oriented design tools, which raise the level of abstraction.

The Agile movement began in the nineties when a number of lightweight frameworks and methods started to appear in software companies (Dingsøyr et al., 2012). They originated from the urge to solve the productivity problem of software development, i.e., the Software Crisis. In 2001, a group of consultants declared that the software development methods and processes at the time focused too much on formal contracts, plans, and documents (Beck et al., 2001; Dingsøyr et al., 2012; Larman, 2004). They perceived that there was an overconfidence in project plans and the ability of foreseeing the future. Project management research has confirmed this view, especially in large projects: "Over budget, over time, over and over again" is Flyvbjerg's (2011, p. 321) short summary of the performance record of large infrastructure projects. Besides infrastructure, high frequency of project fiasco is true in most sectors of society (Cicmil

& Hodgson, 2006). Projects of any kind and size face inherent uncertainty because they project action into a future that cannot be known in advance, only forecasted (Kreiner, 2020). According to Flyvbjerg (2018), the root cause of project failure is that managers keep underestimating scope changes and complexity in project after project. Actually, it is not the scope changes and complexity in themselves that are the main problem, it is how managers overconfidently underestimate these problems (Flyvbjerg, 2018).

A declaration signed in February 2001 (Beck et al., 2001) was named the Agile Manifesto and contained four values and twelve principles describing what *being Agile* is. The essence of the Agile Manifesto (Beck et al., 2001) is the ability to embrace change, rather than believe too much in detailed planning. Also, it was aimed at putting more focus on people and communication rather than plans and formal contracts. The benefits of Agile ways of working have since been proven by scientific researchers and market investigations and comprise an accelerated time to market, increase in quality and productivity, improved IT/business alignment, and enhanced flexibility (Campanelli & Parreiras, 2015; Laanti & Kettunen, 2019; VersionOne, 2020).

When the manifesto was defined, most of the experience originated from applying Agile principles to small teams, and small projects (Laanti, 2012), and the Agile principles were reviewed in 2011 since, at this time, the review board had more experience of implementing Agile ways of working and had also applied them in larger projects and organizations (ibid.). The values are described in this way:

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work, we have come to value: Individuals and interactions over processes and tools; Working software over comprehensive documentation; Customer collaboration over contract negotiation and Responding to change over following a plan. That is, while there is value in the items on the right, we value the items on the left more. (Beck et al., 2001, p. 2)

These values describe what Agile is, but when it comes to understanding what *being Agile* really entails, the twelve principles are more useful. These are the twelve Agile Principles (AP) with some clarifications:

AP1: "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software" (Beck et al., 2001, p. 18). In order to deliver results *early and continuously*, early work such as analysis and design cannot be conducted for a longer period of time, detailing every

part of the requirements or expected results. For software development, *valuable software* means two things, as noted by Bider (2015). First, the software is functioning and should be free from major bugs and, secondly, the software should be used in the context for which it is intended. This means that to be Agile, results must be delivered with high quality and used by someone to add value to the customer no matter the context. Frequently delivering results is not enough on its own.

AP2: "Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage" (Beck et al., 2001, p. 18). To be able to *welcome changing requirements*, one needs to be humble enough to understand that the requirements that will give our customers a *competitive advantage* need to mature over time (Gustavsson, 2011).

AP3: "Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale" (Beck et al., 2001, p. 18). This continuous pulse of work and delivering results is sometimes called a time-box or sprint (Schwaber & Beedle, 2002). Using sprints to become Agile is to be able to break down the intended work into such small parts that valuable results could be made and delivered in less than roughly a month's time.

AP4: "Business people and developers must work together daily throughout the project" (Beck et al., 2001, p. 18). Since Agile work is conducted in short and frequent iterations, it enables higher transparency for the different stakeholders. By applying this principle to be Agile, problems become visible earlier (Cockburn & Highsmith, 2011).

AP5: "Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done" (Beck et al., 2001, p. 18). Although everyone wants motivated people since they are more creative and productive (Amabile & Kramer, 2007), it is the second sentence that is a requirement for being Agile, which hopefully brings more motivated people. Trusting people to make decisions on their own induces autonomy to the team and the single team member (Gupta & Reddy, 2016).

AP6: "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation" (Beck et al., 2001, p. 18). Does this Agile principle mean that a distributed team cannot be Agile? No, but the team needs to live according to the principle by having, e.g., Skype meetings face-to-face or meet regularly in order to be Agile.

AP7: "Working software is the primary measure of progress" (Beck et al., 2001, p. 18). To be Agile, the amount of working, implemented, functionality in the IT system must be the most important measurement to use, more highly prioritized than any other chosen key performance indicator (Cockburn & Highsmith, 2001).

AP8: "Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely" (Beck et al., 2001, p. 18). To achieve a sustainable working pace, it is important to plan and estimate work often and regularly. As described previously for AP3, a commonly adopted way to ensure the regularity is to work in time-boxes or sprints with a set time-span, preferably shorter than a month. To be Agile, teams must be allowed the mandate to decide how much work they accept for a sprint (Hoda & Murugesan, 2016). Thus, they can work a normal number of hours, creating a work environment with *sustainable development*.

AP9: "Continuous attention to technical excellence and good design enhances agility" (Beck et al., 2001, p. 18). As the definition shows, this is not a prerequisite for being Agile but a principle used for enhancing the Agile ways of working.

AP10: "Simplicity - the art of maximizing the amount of work not done - is essential" (Beck et al., 2001, p. 18). As described in AP1, this principle means, that to be Agile, to avoid doing work that might not be of value, such as making detailed plans too far into the future (Cockburn & Highsmith, 2001).

AP11: "The best architectures, requirements, and designs emerge from self-organizing teams" (Beck et al., 2001, p. 18). Self-organizing teams are also sometimes called autonomous teams (Moe et al., 2009). The autonomy in an Agile team can differ, which means that the team could have a different amount of mandate on decisions regarding the end product (goals), people (resources), and planning (Janz, Colquitt, & Noe, 1997). For ASD teams to be Agile, they must be allowed to decide *how* to build the product, while the client decides *what* to build (Agile Alliance, 2001). According to research on Agile autonomous teams, they should be composed of people with a variety of skills in order to manage the variety in their environment, and individuals within the team should, to a high degree, be able to assist and replace each other as required (Stray et al., 2018).

AP12: "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly" (Beck et al.,

2001, p. 18). This routine is sometimes referred to as *retrospective* (Schwaber & Beedle, 2002). To be Agile, it is important to allow the team to reflect and take actions for improvements, as time to reflect will increase team productivity and creativity (Amabile & Kramer, 2007; Williams & Cockburn, 2003).

The Agile Manifesto is a document that is discussed and argued about a lot (Laanti, 2012). Most of the arguments are related to how the Agile Manifesto is understood (or rather, not understood) and whether people agree or do not agree about certain values or principles (Laanti, 2012). To help in understanding and interpreting the document, a non-profit organization called the Agile Alliance (Agile Alliance, 2001) was formed in the same year as the Agile Manifesto (Beck et al., 2001) was signed.

Does this mean that an organization needs to abide by the four values and all twelve principles to call themselves Agile? According to Cockburn & Highsmith (2001), all principles are needed in order to claim to be Agile since the manifesto was not intended for cherry-picking, but Kettunen (2009) claims that there is no uniform definition of ASD and instead provides a comprehensive list of different definitions of ASD. Laanti (2012) expands Kettunen's (2009) list and presents different definitions of what *being Agile* means as described by several practitioners and researchers. Her conclusion is that the Agile literature defines no point after which an organization has adopted enough practices to be called Agile. Instead, the literature presents various routines, practices, and mindsets with which to approach different kinds of problems. She refers to Appelo (2011), who claims that most people have misunderstood agility since they have not understood complexity theories, which Agile originates from. Complexity theory means that any simplistic, linear model is bound to fail and that the success of an organization *being Agile* comes from focusing on adaptability, not predictability (Appelo, 2011). Therefore, it would be wrong to say that an organization is *being Agile* by ticking off several items on a list. Because of this complexity, some routines might be needed in one organization but not in another to become Agile.

The Agile Alliance (2001) defines Agile in the following way which will be the definition used in this thesis: *"Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development and the 12 Principles behind it"*.

The Agile Alliance (2001) also pinpoints how this way of working differs from other approaches:

> One thing that separates Agile from other approaches to software development is the focus on the people doing the work and how they work together. Solutions evolve through collaboration between self-organizing cross-functional teams utilizing the appropriate practices for their context. (Agile Alliance, 2001)

To clarify this definition, this means that Agile is not only about a number of practices. It is neither only about teamwork nor only about responding to change. These are typical misconceptions of what Agile is and has caused *shallow adoptions* of ASD (Gregory et al., 2015), where organizations have adopted only one part of the definition. Agile is rather a systematic approach incorporating the Agile values and principles with a focus on the people working in self-organizing, cross-functional teams.

A number of consultants have developed methods and frameworks based on the Agile values and principles described in this section. Some of these consultants even signed the Agile Manifesto (Beck et al., 2001). In the next section, I will present one them: the Scrum framework.

2.2 The Scrum framework

Presently, the most widely used Agile software development framework by far is Scrum (Laanti & Kettunen, 2019; VersionOne, 2020). Scrum was also used by the individual teams in all cases described in this thesis and will, therefore, be briefly described in this section. I will focus on some important terms, concepts, and specifically, roles, as they are described in The Scrum Guide (Sutherland & Schwaber, 2013). The role titles from Scrum are often used regardless of which practices or methods that are adopted (Gustavsson, 2017) and are therefore important to understand when investigating large-scale ASD. The Scrum Guide (Sutherland & Schwaber, 2013) is an online document explaining Scrum by the original method-makers and is sometimes updated based on feedback by Scrum practitioners.

Scrum is an iterative and incremental Agile framework originally intended for small, independent software development teams (Schwaber & Beedle, 2002). Software is developed during two- to four-week-long software development iterations, which are called *sprints*. A sprint starts with s*print planning,* which is a collaborative routine to estimate and decide what can be delivered in the sprint and how this work will be achieved. Each day contains a short routine within the team called the *Daily Scrum*, where each team member presents progress, coordinates work, and solve problems. The sprint ends with a *review* (sometimes called *demo*) to receive

feedback from stakeholders, as well as a *retrospective* where the teams decide on improvements for the next sprint.

2.2.1 Roles and team autonomy

The original Scrum roles were defined in 1993 by Jeff Sutherland and were named *Team*, *Scrum Master*, and *Product Owner* (Sutherland & Schwaber, 2011). The *team*, which should be self-organized and cross-functional, consists of developers that have the competence and capability of implementing and delivering any requirements imagined for the software. The Scrum Guide (Sutherland & Schwaber, 2013) stresses the importance of cross-functionality, meaning that specific roles in the team, such as tester or analyst, does not need to be appointed due to the intended cross-functionality. In other words: everyone does what is needed to complete the job, and the team decides who does what.

What team autonomy really means is that the team should be allowed to decide *how* best to accomplish their work (Sutherland & Schwaber, 2013). This means that the team should be allowed to decide on, e.g., who does what, which tools to be used for testing and when to have meetings within the team. Team autonomy in ASD does not, however, mean that the teams can decide *what* to build. Although this sounds simple enough, the process of forming and implementing teams with high autonomy is still not fully understood in the context of complex team-based, knowledge-intensive organizations (Mathieu et al., 2008; Moe et al., 2019).

A *Scrum Master* guides the team in the implementation of the Scrum practices, protects the development team from external disruptions and tries to facilitate and improve the Agile ways of working in the Scrum team (Sutherland & Schwaber, 2013). A Scrum Master is a servant-leader for the team. Since Agile teams are self-organizing, a Scrum Master does not have the mandate to decide how the team should work or who should do what. The Scrum Master facilitates meetings, ensures that the meetings take place, that attendees understand the purpose, but does not have any more mandate in decisions than a team member. The Scrum Master helps those outside the team understand which of their interactions with the team are helpful and which are not, and helps everyone change these interactions to maximize the value created (Sutherland & Schwaber, 2013).

A *Product Owner* is responsible for maximizing the value of the product resulting from the team's work by creating and prioritizing the requirements for the software. He or she is also responsible for guiding the team by clarifying needs and business goals in implementation of features that

can meet these requirements. In other words, the Product Owner decides *what* to build, while the team decides *how* the work is done. For the Product Owner to succeed, the entire organization must respect his or her decisions. The decisions of the Product Owner are visible in the content and order (based on priority) of the Product Backlog. No one can force the team to work from a different set of requirements (Sutherland & Schwaber, 2013).

2.2.2 Coordination according to Scrum

Since self-organizing is an important principle in ASD, the basic principle is that the teams should work out dependencies or other issues with the surrounding organization themselves, without adding additional roles besides the Scrum Master and the Product Owner. The Product Owner should have enough mandate to help the team on issues concerning *what* to build, such as deciding on conflicting requirements. Because the team and the Product Owner is responsible for defining and improving coordination practices, Agile can be understood as a bottom-up approach to coordination (Moe et al. 2018).

In a project with several Scrum teams, Schwaber (2004) added an additional inter-team coordination routine called *Scrum of Scrums* in the Scrum framework. Scrum of Scrums was originally described in a case study report by Sutherland (2001), who introduced the Scrum framework at the company IDX Systems in 1996 (now GE Healthcare), where hundreds of developers worked on dozens of products. Scrum of Scrums was originally described as a weekly meeting between all Scrum Masters in a product line to discuss and solve dependency issues between teams. Scrum of Scrums adopted in this form, as weekly meetings, has been widely criticized (Paasivaara et al., 2012). When only Scrum Masters are allowed in the meetings, there is a risk that solutions are merely discussed and never carried out. Practitioners have since stressed that the Scrum of Scrums should be a place for solving coordination issues, not a meeting for Scrum Masters, and put forth the need for allowing any team member with a coordination issue to participate in the Scrum of Scrums (Paasivaara et al., 2012).

Another criticism of the Scrum of Scrums routine has been that a weekly meeting does not solve problems between teams fast enough. Since the Scrum teams meet on a daily basis to sort out problems, which is called the *Daily Scrum*, having to wait for a whole week to sort out problems between teams is seen as much too slow (Cohn, 2007; Paasivaara et

al., 2012; Schwaber 2004). Hence, practitioners have argued for more frequent Scrum of Scrums, either 2-3 times a week (Cohn, 2007) or daily (Schwaber, 2004).

Sutherland (2001) did not give any details on how the original Scrum of Scrums were conducted at IDX Systems, but Schwaber (2004) suggests that the meeting should use the same agenda as the *Daily Scrum*, by letting attendees answer three questions: 1) What have we done since the last meeting? 2) What are we doing until the next meeting? 3) Are there impediments to get there?

Schwaber (2004) also suggests using a fixed meeting time, time-boxed to last a maximum of 15 minutes. Cohn (2007), on the other hand, suggests allocating 30 to 60 minutes for Scrum of Scrums to let people discuss and solve problems that might arise at the meeting while the people involved are present.

Even though Scrum of Scrums was initially presented by both Sutherland (2001) and Schwaber (2004), the inter-team coordination routine is not described in The Scrum Guide (Sutherland & Schwaber, 2013) which only consists of descriptions of how to work with one single Scrum team.

If the organization fails to solve dependency or coordination issues fast enough on its own, Schwaber (2004) acknowledges that additional roles might be useful besides the Product Owner, Scrum Master, and the Team. One such role in the organization is, of course, the *manager*, and according to Sarpiri and Gandomani (2017), the main responsibility for the manager in an ASD organization using Scrum is to facilitate the Agile ways of working as well as handling problems and challenges faced by the teams. Shastri, Hoda, and Amor (2017) also describe that managers need to work as mentors, coordinators, negotiators, and process adapters in an Agile organization using Scrum.

From this presentation of small, independent Scrum teams, in the next section we move on to discuss projects where several teams work together according to SAFe.

2.3 The Scaled Agile Framework (SAFe)

In the past few years, several Agile frameworks designed for large-scale settings have become widespread, but with diverse views and different prescribed routines on how to manage coordination. According to the industry study from the company VersionOne Inc. (2020), the most adopted of the new frameworks is the Scaled Agile Framework (SAFe),

which is used by 35 percent of the respondents. Each other large-scale Agile framework was only adopted by, at most, four percent of the respondents in the survey. This means that SAFe is, by far, the most common large-scale Agile framework, and the number of organizations adopting SAFe has increased annually compared to other frameworks (VersionOne, 2020). Although there are other large-scale Agile frameworks, such as LeSS (Larman & Vodde, 2010), DAD (Ambler & Lines, 2012), or the Spotify Model (Salameh & Bass, 2020), I will not present any other since the routines described in SAFe were implemented in all three case organizations investigated in this study. Also, several inter-team coordination routines presented in this chapter are present and explained in the other large-scale frameworks although sometimes with a different name. E.g. the Scrum of Scrums routine described in SAFe (Scaled Agile, 2020), has the same name and is described in the same way in LeSS (Larman & Vodde, 2010).

In short, SAFe is a framework that focuses on applying Agile methodologies to the whole organization. SAFe has routines that cover the *Portfolio level* responsible for investments, *Program level* responsible for the execution of the planned initiatives, and *Team level* responsible for the work and coordination in and between teams (Scaled Agile, 2020). Teams are put together with a mix of competencies such as developers, testers, and business developers to become cross-functional (Scaled Agile, 2020). Mixing competencies in each team is a way to fulfill the fourth value of the Agile Manifesto (Beck et al., 2001, p. 18): "Business people and developers must work together daily throughout the project".

On Portfolio and Program level, SAFe builds on Lean values and principles of *product development flow* (Reinertsten, 2009), which means identifying and improving steps in the value stream for products (or services) and eliminate steps that do not create value.

On Team level, SAFe provides a structured way of organizing the work with planning days for all teams, and other routines for inter-team coordination. The name for the set of teams who plan and cooperate together is an *Agile Release Train*. In the following subsections, the prescribed inter-team coordination routines of SAFe will be explained from a team-level perspective divided into the same five areas of coordination as described in Okhuysen and Bechky (2009) and also used in Gustavsson (2018).

2.3.1 Plans and rules

The SAFe framework prescribes the use of a plan for a period of time called a Program Increment (PI) spanning from eight to twelve weeks. A PI consists of four time-boxed iterations (or *sprints*) where the teams in the Agile Release Train demonstrate results at the end of each sprint to receive feedback to adjust their work. This routine is designed in accordance with the twelfth principle in the Agile Manifesto (Beck et al., 2001, p. 18): "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly." Besides reflections, the repeating attention to review performed work at the end of each sprint is in line with the ninth principle in the Agile Manifesto (Beck et al., 2001, p. 18): "Continuous attention to technical excellence and good design enhances agility."

Also, the setup of having four sprints during a PI is designed based on the second principle in the Agile Manifesto (Beck et al., 2001, p. 18): "Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage". It is also intended as a way to balance one of the values expressed in the Agile Manifesto (Beck et al., 2001, p. 2): "Responding to change over following a plan."

Another reason for deciding on the exact content of work being done for a whole PI is to be clear on what should *not* be done during the PI. This intended goal is in line with the tenth principle of the Agile Manifesto (Beck et al., 2001, p. 18): "Simplicity – the art of maximizing the amount of work not done – is essential."

At the end of a PI, a two-day PI planning event is conducted to create a new plan for the following PI. The PI planning is, among practitioners, what SAFe is most known for. The routine is so essential to SAFe that "If you are not doing it, you are not doing SAFe" according to the framework originators (Scaled Agile, 2020).

Figure 2. Agenda for the PI planning event (Source: Scaled Agile, 2020).

The PI planning event follows a detailed standardized agenda, see Figure 2, created in order to involve all team members as well as stakeholders in creating and committing to a detailed plan. By making a realistic plan based on previous data of estimates and actual results, the PI plan is designed to follow the eighth principle of the Agile Manifesto (Beck et al., 2001, p. 18): "Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely".

2.3.2 Objects and representation

SAFe uses the two objects *product backlog* and *sprint backlog* to visualize upcoming work. The product backlog is *one* prioritized list that defines all of the work to be done on a product (Larman & Vodde 2010) for all teams and is a *boundary object*. A boundary object is "a sort of arrangement that allow different groups to work together without consensus" (Leigh Star 2010, p. 602). Items in the list are described in different detail, depending on how prioritized they are. The prioritization is based on the first principle of the Agile Manifesto (Beck et al., 2001, p. 18): "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."

The items at the bottom of the list, the least prioritized, are often expressed as business objectives (sometimes called *epics*) or vaguely described features of the IT system. Most of them are described by businesspeople using their own terminology. Progress is measured based on the product backlog in accordance with the seventh principle in the Agile

Manifesto (Beck et al., 2001, p. 18): "Working software is the primary measure of progress."

To be able to create a product backlog, a *program vision*, and a *roadmap* (containing milestones and important events) are often used as a basis to guide the work (Scaled Agile, 2020).

As items are being prioritized, developers and businesspeople refine, split, and add more details to the items. Therefore, items at the top of the list are written and understood by both parties and can be divided into small work packages (sometimes called *stories*) in sprint backlogs, which are one list per team of committed work for the upcoming sprint of two to four weeks. The sprint backlog is thereby *not* a boundary object; it is a clearly defined and prioritized list of work fully understood by the development team but maybe not to businesspeople (depending on their technical knowledge). The strategy of performing work in short sprints is based on the third principle of the Agile Manifesto (Beck et al., 2001, p. 18): "Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."

The recommendation from SAFe is to display the sprint backlog on a wall, called a Scrum board or Kanban, using post-it stickers to describe tasks and columns on the wall to visualize the status of the tasks with column names such as "not started", "started" and "done" (Scaled Agile, 2020).

Since there might be dependencies between sprint backlogs between teams, the SAFe framework prescribes a visual representation called a *program board*.

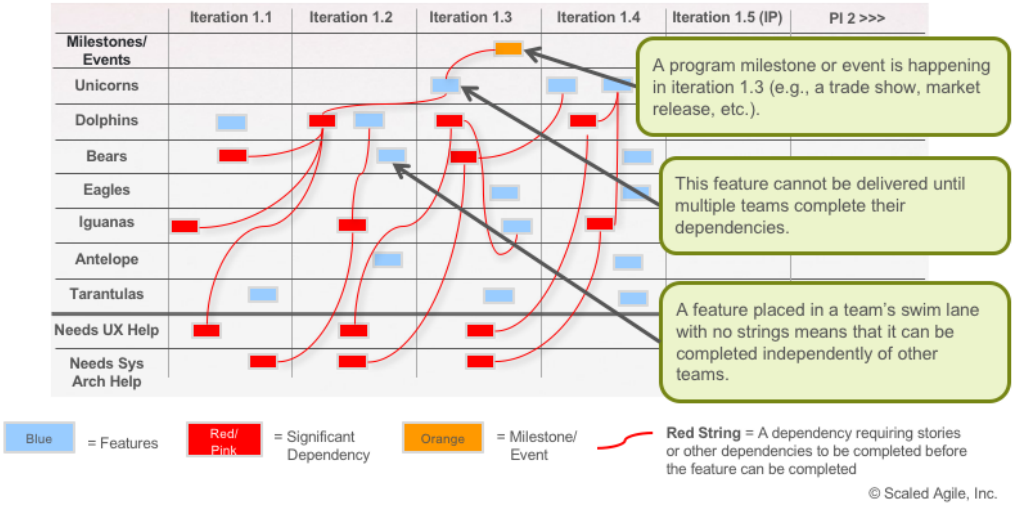

Figure 3. Example of program board (Source: Scaled Agile, 2020).

A program board shows the different teams involved in the upcoming release on the Y-axis and the features planned in the sprint backlogs along the X-axis. Dependencies between stories are shown by using red strings, as shown in Figure 3. By implementing routines for managing the program board, the current situation of dependencies can be visualized and kept track of (Gustavsson, 2020).

2.3.3 Roles

As presented previously, SAFe stresses the importance of creating cross-functional, self-organizing teams with people of complementing competences. This strategy is based on both the fifth principle of the Agile Manifesto (Beck et al., 2001, p. 18): "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done", as well as the eleventh principle (Beck et al., 2001, p. 18): "The best architectures, requirements, and designs emerge from self-organizing teams."

SAFe prescribes the use of the roles Scrum Master and Product Owner, originally invented in Scrum (Schwaber & Beedle, 2002). Regarding the Scrum Master, SAFe puts forth that horizontal inter-team coordination is a responsibility for the Scrum Master but cannot be delegated entirely to this role since the team members should share the responsibility as well (Scaled Agile, 2020).

SAFe (Leffingwell, 2007) prescribes one Product Owner per team, but since additional supporting roles are critical in large-scale projects for managing the growth of inter-dependencies (Blichfeldt & Eskeröd, 2008), more roles are suggested (Scaled Agile, 2020). One such role is the *Product Manager* who coordinates Product Owners and other stakeholders to create the roadmap and product backlog (Scaled Agile, 2020). Another coordinating role is the *System Architect*, responsible for defining and communicating an architectural vision for the entire Agile Release Train (Scaled Agile, 2020).

To coordinate the entire Agile Release Train along with the Product Manager and System Architect, an additional role is prescribed named Release Train Engineer. The Release Train Engineer is a servant leader and coach who facilitates events and routines and assists the teams in delivering value (Scaled Agile, 2020). Hence, the Release Train Engineer could also be called a "Chief Product Owner" (Stojanov et al., 2015) or "Chief Scrum Master" (Leffingwell et al., 2017).

SAFe suggests several additional roles on the Portfolio and Program level, but they are not presented here because the interest of this study is on the *Team level*.

2.3.4 Proximity

SAFe (Scaled Agile, 2020) prescribes inter-team coordination routines to be conducted face-to-face, in accordance with the sixth principle in the Agile Manifesto (Beck et al., 2001, p. 18): "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."

Already the early Agile methods such as Scrum and XP stressed the importance of co-location for the team. Nyrud and Stray (2017) observed that informal and ad hoc conversations emerged in large-scale ASD projects because of teams being co-located in one office building. However, that does not mean that benefits arise only because of the co-location. It also has to do with architecture and office rules. One increasingly popular way of structuring office space is to create an 'open' office architecture where people can sit wherever they like. Interestingly, a recent study showed that face-to-face communication between employees in 'open' office locations *decreased* by approximately 70 percent (Bernstein & Turban, 2018). Instead of increasing face-to-face collaboration, the 'open' office appeared to trigger a response to socially withdraw from each other and instead turn to virtual communication. This shows the importance of co-located *teams*, not just co-located employees, without being close to your team members.

Although being co-located is an enabler for both scheduled and unscheduled meetings and forums, there might be practical problems such as put forth in Šmite et al. (2017), who found that it was difficult to have unplanned meetings because of too many scheduled meetings and a lack of meeting rooms.

SAFe (Scaled Agile, 2020) did not prescribe or suggest any kind of specific advice on what to do in distributed, or virtual, organizational settings before the COVID-19 virus became a global pandemic. It only put forth that the best solution is to have all teams co-located, thereby decreasing the risk of coordination problems. Also, descriptions of face-to-face meetings and objects such as the program board using physical strings assume a co-located work area. During the spring of 2020, however, some articles appeared on the SAFe website (Scaled Agile, 2020),

where companies shared their experiences of conducting inter-team coordination routines in distributed ways.

2.3.5 Routines

Typically, routines in all kinds of projects are meetings, either planned or unplanned. Collocated meetings are effective because physical proximity allows richer communication, which enables swifter and more flexible coordination (Groth, 1999). Dietrich et al. (2013) also point to prior studies which showed that technological novelty often relates to a higher rate of group meetings instituted by management. However, Okhuysen and Bechky (2009) also present that unplanned contingencies and responses to them are important for the efficiency of workflow and processes in organizations. Consequently, both routines for planned and unplanned coordination are important in large-scale ASD projects (Moe et al., 2018).

SAFe prescribes performing the routines invented by Scrum within the team, such as *Sprint planning* and *Retrospectives* (Schwaber & Beedle, 2002). For the *demo* (or *review*) intended for the single team and stakeholders in Scrum (Sutherland & Schwaber, 2013), SAFe prescribes a *system demo*, which instead means that all teams demonstrate their results together to get feedback from stakeholders simultaneously (Scaled Agile, 2020).

For routines on planned coordination, PI planning was presented above. For unplanned coordination, SAFe prescribes two routines: Scrum of Scrums and Communities of Practice (Scaled Agile, 2020).

Scrum of Scrums is a recurring meeting with an agenda similar to the *Daily Scrum*, which helps to coordinate dependencies between teams and provides insight into progress and impediments. SAFe prescribes Scrum of Scrums to be performed two to three times per week and divided into two parts: 1) Scrum of Scrums for everybody attending and 2) Meet-after/Problem-solving for representatives that need to sort out dependency issues.

SAFe also prescribes that the Scrum Master should be the team representative, while other practitioners (e. g. Larman & Vodde, 2010) suggest any team member, maybe on a rotating schedule, could represent the team in the meeting. Compared to the *Daily Scrum*, SAFe suggests adaptations of the agenda, focusing on what issues are *of importance to the other teams* regarding what has been done, what will be done, and obstacles to reach the current sprint goal (Scaled Agile, 2020). However, Scrum of Scrums has been found to be problematic in larger projects according to several studies (Paasivaara et al., 2012; Rolland, Mikkelsen, & Næss, 2016).

Regarding Communities of Practice, the concept is described in the SAFe framework as organized groups of people who have a common interest in a specific technical or business domain who regularly collaborate to share information, improve their skills, and actively work to advance the general knowledge of the domain (Leffingwell et al., 2017). The intended purpose is to coordinate and disseminate knowledge in different domains on a voluntary basis (Scaled Agile, 2020). Several examples of organizing Communities of Practice are also described in SAFe such as role-based (for Product Owners, Architects or Testers) or topic-based (for java-development or automated testing). In comparison with other routines in SAFe, there are no prescribed details on how to perform meetings or other events within Communities of Practice.

Besides Scrum of Scrums and Communities of Practice, SAFe puts much effort into emphasizing the importance of routines for *cadence* and *synchronization*. Therefore, routines for developing in sprints are prescribed to always consist of two weeks of development for all sprints (for all teams), ending in integration and demonstration to synchronize results between teams. Also, above the Team level, managers are coordinated through weekly meetings with senior management, but no details are described for these kinds of meetings (Scaled Agile, 2020).

As can be seen, all twelve principles of the Agile Manifesto (Beck et al., 2001) are addressed in SAFe. Despite this, critique has been put forth about how Agile SAFe really is. This critique will be presented in the next section.

2.3.6 Critique against SAFe

SAFe has been implemented in many organizations (e.g., PepsiCo, Cisco, Telia, Philips, AstraZeneca, and Intel), and early adopters have reported significant improvement in terms of productivity, quality, time-to-market, employee engagement and job satisfaction (Scaled Agile, 2020). Nevertheless, there is critique against implementing SAFe.

The routines in SAFe are described and recommended in detail, which might leave less flexibility for spontaneous meetings to emerge and for teams to take the initiative for coordination (Moe et al., 2018). Both researchers (Alqudah & Razali, 2016; Ebert & Paasivaara, 2017, Stojanov et al., 2015) and Agile practitioners (Schwaber, 2013; Maximini, 2013) have criticized SAFe for being too strict and formal, based on detailed prescriptions in the framework. Being too prescriptive also risks constraining the continuous improvement of the teams (Alqudah & Razali, 2016).

The framework itself does not give any guidance on tailoring. This may not be surprising, though, since few software development methods contain advice to aid tailoring (Iivari, 1989). However, It might seem like a contradiction that even though a fundamental value of *Agile* software development is to adapt and improve continuously, SAFe very seldom encourages any kind of tailoring. Instead, SAFe presents a static picture of how to conduct routines and does not give any guidance as to whether some measures are more important at the start of the implementation or whether some are more important later on.

2.4 Research on large-scale ASD

First, a definition of what *large-scale* means in large-scale ASD. Participants of the XP2014 large-scale Agile workshop gave varying definitions for large-scale Agile development (Dingsøyr & Moe, 2014), showing that what is seen as large-scale depends very much on the context and the person defining it. Dingsøyr et al. (2014) summarized different definitions and the use of the term *large-scale Agile* as a result of a literature review. Size has been previously described in terms of size in several people or teams, project budget, codebase size, and project duration. Instead of taking all these terms into account, Dingsøyr et al. (2014) simplified matters by defining large-scale by the number of collaborating and coordinating teams. Their proposed categorization is to define two to nine collaborating teams as "large-scale" and more than ten collaborating teams as "very large-scale."

Dikert, Paasivaara, and Lassenius (2016) had the same view and identified several additional studies that discussed interpretations of large-scale. The authors put forth that most of these studies referred to the number of people involved in defining what large-scale is without other kinds of complexities.

In the early work on Agile methods, Fowler (2000) considered the Crystal methodology to be suitable for up to 50 people. The same number of people has been reported by practitioners and researchers to be the maximum group size suitable for Agile ways of working (Williams & Cockburn, 2003). Other studies have referred to Agile projects including up to 50 people as small (Koehnemann & Coats, 2009), and considered a development project large if it incorporated between 50 and 100 people, including all project personnel (Elshamy & Elssamadisy, 2006).

I will use the definition proposed by Dikert et al. (2016) of large-scale, which is: *a software development organization with 50 or more people or at least six teams*. Having six teams with an average size of six to seven people, plus some supporting staff, can reasonably be considered to form an organization close to 50 people or more. The definition by Dikert et al. (2016) includes both organizations that as a whole focus on software development, as well as the parts of larger (non-software focused) organizations that develop software, e.g., in-house software development units of large non-software development corporations. This means that the *software development organization* does not need to consist of only developers, but enough many people and teams must belong to the same software development organization developing a common product or project, and thus have a need to collaborate. I argue that the definition suggested by Dingsøyr et al. (2014), which defines large-scale already when two teams cooperate, is problematic since when only a few teams are cooperating, an overview of the whole project, and dependency issues might be easier to obtain even without additional routines. Hence, large-scale inter-team coordination routines might not be implemented in Agile projects with only two or three teams, thereby making it less complex and without the challenges associated with large-scale ASD.

2.4.1 Challenges in large-scale ASD

In large-scale ASD projects, there is a need for extensive and dynamic knowledge exchange between and within teams to solve problems and adjust for emerging changes (Hoegl et al., 2004). Scrum and other early Agile frameworks were designed without much consideration for these scaling problems with a focus on solving coordination problems within the team. The originally proposed way to scale up an Agile organization was to employ multiple small Scrum teams that simultaneously developed different parts of the same software system (Schwaber, 2007). Unfortunately, there are usually architectural complexities that result in a network of dependencies between requirements (Benestad & Hannay, 2011; Carlshamre, 2002), which increases the difficulty of coordination.

In his doctoral thesis, Heikkilä (2015) focuses on release planning practices and presents results from the case study performed at the company F-Secure. The author describes: "release plans also showed what dependencies the teams had with each other and when the teams were planning to initiate or finish developing the dependent features, which helped in the dependency management." (Heikkilä, 2015, p. 48). The au-

thor further describes: "dependencies between features are often complex and difficult to identify before implementation. Consequently, release planning methods should not expect that all dependencies are known beforehand" (Heikkilä, 2015, p. 70).

In the same case study, the author also describes a challenge within the case organization, which was that many developers expected that "someone else" would manage the coordination between the teams while the Agile framework expects developers to coordinate and manage dependencies themselves (Heikkilä, 2015). This might be due to the legacy of older ways of working, since there is a difference between traditional project management approaches and Agile ways of working. Traditional project management approaches typically focus on formal coordinating mechanisms through a pre-defined process, precise and in-depth documentation, and high levels of specialization in role assignments (Nerur et al., 2005). Agile approaches, on the other hand, tend to favor self-management, i.e., teams determine the best way to handle work (Gustavsson, 2019a) and emerging processes rather than following a predetermined workflow (Gustavsson, 2019c). They also favor more informal coordinating mechanisms (Boehm & Turner, 2005). According to Moe et al. (2019), *mutual adjustments* (Thomson, 1976) tend to be the primary coordination mechanism used in agile teams. However, Agile projects involve specific challenges related to coordinating and communicating with multiple stakeholders, as Agile development often require frequent releases. Collaboration with customers (Nuotilla, Aaltonen, & Kujala, 2016) and, as put forth by Xu (2009), a more informal approach to coordination can sometimes become challenging.

Several problems related to Agile development in large-scale contexts have also been identified by van Waardenburg and van Vliet (2013). These included a multi-layered organizational structure with strict communication channels, diffused information, dependencies on external parties, difficulties in developing end-to-end functionality, lack of business involvement, and requirements prioritization when many stakeholders with conflicting needs are involved. Most of these problems could be considered coordination issues, especially coordination between different teams residing in different departments or even companies. Hence, challenges with inter-team coordination in large-scale ASD and how they have been resolved is discussed in the next section.

2.4.2 Challenges and solutions of inter-team coordination

Because of the lack of support for managing coordination issues in large-scale Agile settings in the first Agile frameworks, many diverse solutions have been created to handle different aspects of coordination problems.

Jarzabkowski et al. (2012) suggest that new coordination routines are gradually established through existing social routines. The authors argue that new routines evolve in the organization. So, for example, in a large-scale Agile project only using Scrum of Scrums to coordinate activities, new forms of coordination will emerge when the organization discovers the need for coordination between other activities or actors, such as between Product Owners. Arguably, this could happen in large-scale Agile projects that start off with simple coordinating mechanisms from Scrum, such as daily meetings and sprint planning, only to discover that large-scale projects often require additional routines for coordinating, for example, architecture meetings across teams and reviews/demonstrations involving several teams working on the same functionality.

Although this internal invention of coordination routines may occur in some organizations, many organizations implement routines that are already in use in the industry (Laanti & Kettunen, 2019; Moe et al., 2016; VersionOne, 2020).

2.5 Impacts on teams in large-scale ASD

To understand what we know about what large-scale ASD does to software development teams, a systematic literature review (SLR) was conducted to gather results from previous research on perceived impacts. There are several SLRs conducted on success factors and challenges (for example, Dikert et al., 2016), but since no papers have been found specifically addressing impacts on teams, an SLR was deemed appropriate to conduct. Table 1 shows the 27 selected studies from the SLR.

Table 1. SLR publication details

P#	Authors	Year	Title	Publication
P1	Babinet, E., & Ramanathan, R.	2008	Dependency Management in a Large Agile Environment.	Conference
P2	Evbota, F., Knauss, E., & Sandberg, A.	2016	Scaling up the planning game: Collaboration challenges in large-scale Agile product development.	Conference
P3	Gupta, R. K., & Reddy, P. M.	2016	Adapting Agile in a globally distributed software development.	Conference

P4	Hobbs, B., & Petit, Y.	2017	Agile Methods on Large Projects in Large Organizations.	Journal
P5	Laanti, M., Salo, O., & Abrahamsson, P.	2011	Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on Agile transformation.	Journal
P6	Lagerberg, L., Skude, T., Emanuelsson, P., Sandahl, K., & Stahl, D.	2013	The impact of Agile principles and practices on large-scale software development projects: A multiple-case study of two projects at Ericsson.	Conference
P7	Lowery, M., & Evans, M.	2007	Scaling Product Ownership.	Conference
P8	Martini, A., Pareto, L., & Bosch, J.	2013	Improving business success by managing interactions among Agile teams in large organizations.	Conference
P9	Moore, E.	2009	Influence of Large-Scale Organization Structures on Leadership Behaviors.	Conference
P10	Nyrud, H. & Stray, V.	2017	Inter-team coordination mechanisms in large-scale Agile.	Conference
P11	Olszewska, M., Heidenberg, J., Weijola, M., Mikkonen, K., & Porres, I.	2016	Quantitatively measuring a large-scale Agile transformation.	Journal
P12	Paasivaara, M., Durasiewicz, S., & Lassenius, C.	2008	Distributed Agile development: Using Scrum in a large project.	Conference
P13	Paasivaara, M., & Lassenius, C.	2016	Scaling Scrum in a large globally distributed organization: A case study.	Conference
P14	Paasivaara, M., Lassenius, C., & Heikkilä, V. T.	2012	Inter-team coordination in large-scale globally distributed Scrum: Do scrum-of-scrums really work?	Conference
P15	Rolland, K. H., Mikkelsen, V., & Næss. A.	2016	Tailoring Agile in the large: Experience and reflections from a large-scale Agile software development project.	Conference
P16	Sekitoleko, N., Evbota, F., Knauss, E., Sandberg, A., Chaudron, M., & Olsson, H. H.	2014	Technical dependency challenges in large-scale Agile software development.	Conference
P17	Wigander, J.	2014	Experiences from implementing Agile ways of working in large-scale system development.	Book chapter
P18	Dingsøyr, T., Moe, N. B., Fægri, T. E., & Seim, E. A.	2018	Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation	Journal

P19	Gupta, R. K., Jain, S., & Singh, B.	2018	Challenges in Scaling up a Globally Distributed Legacy Product	Conference
P20	Laanti, M. & Kettunen, P.	2019	SAFe Adoptions in Finland: A Survey Research	Conference
P21	Paasivaara, M.	2017	Adopting SAFe to Scale Agile in a Globally Distributed Organization	Conference
P22	Paasivaara, M., Behm, B., Lassenius, C., & Hallikainen, M.	2018	Large-scale agile transformation at Ericsson: a case study	Journal
P23	Pries-Heje, J., & Krohn, M. M.	2017	The SAFe way to the agile organization.	Conference
P24	Salameh, A., & Bass, J.	2019	Spotify Tailoring for B2B Product Development	Conference
P25	Schnitter, J., & Mackert, O.	2011	Large-Scale Agile Software Development at SAP AG	Conference
P26	Šmite, D., Moe, N. B., Levinta, G., & Floryan, M.	2019	Spotify Guilds: How to Succeed With Knowledge Sharing in Large-Scale Agile Organizations	Journal
P27	Uludağ, Ö., Kleehaus, M., Dreymann, N., Kabelin, C., & Matthes, F.	2019	Investigating the Adoption and Application of Large-Scale Scrum at a German Automobile Manufacturer	Conference

The first part of Table 1 (P1 to P17) is the result of the first round of searches and ordered alphabetically according to the first authors family name. The second part (P18 to P27) is from the second round of searches and ordered in the same manner.

Most studies were reported from very large companies, some with more than a hundred thousand employees, but that does not necessarily mean that the ASD organization in the company is large. In four studies, there was no direct indication of the size of the organization, but the descriptions of roles and ways of working revealed that the case organization was a large-scale case according to my chosen definition (*50 or more people or at least six teams*).

Where described, the size of the ASD organizations varied from 50 to more than 18 000 people. In twelve studies, the size of the ASD organization was presented in terms of teams, ranging from six to 55 teams. Eleven studies were performed within telecommunications, including companies such as Ericsson, Nokia, and Siemens. Six studies were from software vendors and two studies from cloud software service companies (Salesforce and Spotify). Four studies were from public organizations, of which two were government agencies, and one was the public service broadcaster BBC. Three studies were from the financial sector, where one

was a Norwegian bank and two financial services companies (one of which was SimCorp). Three studies were from the automotive sector, of which one was an automotive manufacturer in Germany. One survey study was performed with a large number of Finnish organizations that had implemented SAFe.

The context descriptions of the selected studies in Appendix D summarize inter-team coordination routines or large-scale ASD frameworks implemented, as well as industries, company names, ASD organization sizes, and study types.

In several cases, there were, unfortunately, no descriptions of the chosen practices, routines, or frameworks implemented in the organizations. In ten of the studies, the Scrum framework was implemented together with the Scrum of Scrums routine for inter-team coordination. Four of the organizations worked according to the framework Large-Scale Scrum (LeSS), of which one used a combination of LeSS and Nexus. Three organizations had implemented SAFe, and two had implemented the Spotify model (of which one of the organizations was actually Spotify). It is somewhat surprising that, although SAFe is the most commonly implemented large-scale ASD framework (according to the latest annual survey by VersionOne, 2020), only three of the organizations in the identified papers worked according to SAFe.

The perceived benefits and drawbacks are organized in a model of higher-order themes and summarized in Table 2. The performed analysis is presented in the research methods chapter (5). The themes are presented individually in the subsequent sections.

Table 2. Model of higher-order themes.

Impact	Theme	Sources	# of coded quotes
Benefit	Coordination and cooperation	P1*2, P3, P6, P10*2, P12, P15*2, P18, P20, P24*2, P26	14
Benefit	Overview and transparency	P1*2, P5, P6*2, P7, P16, P20, P22, P23, P24*2, P25	13
Benefit	Speed, focus, and efficiency	P5*2, P6, P11*4, P13, P20	9
Benefit	Improved quality	P7*2, P11, P14, P15, P17	6
Benefit	Employee motivation	P3, P12, P17, P21, P23, P26	6
Drawback	Routines not helpful for teams	P2, P13*5, P16, P18*2, P19, P20, P21, P22	13
Drawback	Conflicts between teams	P1*2, P2, P5, P7, P8, P9, P10, P16	9
Drawback	Stress	P2, P4, P6, P14, P17*2, P22, P25*2	9
Drawback	Limited autonomy	P4*2, P7, P17, P18, P21	6
Drawback	Unclear responsibilities	P19, P20, P22*2, P23	5
Drawback	More difficult to prioritize	P2, P7*2, P8, P25	5
Drawback	Selfishness	P13, P14, P16	3
Drawback	Too much time spent in meetings	P21, P27	2
Drawback	Shyness	P16, P17	2
Drawback	Clashes with surrounding organizations	P22, P23	2

The described benefits and drawbacks are related to how work was perceived before implementing the large-scale inter-team coordination routines. The comparison is *not* with how work was perceived before Agile ways of working was implemented, since ASD had been implemented for several years in all studied organizations.

2.5.1 Coordination and cooperation.

Several examples of increased coordination and cooperation were presented in the studies. Open communication [P1] and broad, collaborative discussions [P1] were prominent at Salesforce:

> This approach leads attendees to take responsibility for the content of the meeting and has resulted in more productive and collaborative discussions… [These] practices … are all designed to open communication and promote collaboration. [P1]

In one case, one practice was to let one expert in the team share knowledge with other team members regularly: "This practice has a huge impact on distributing knowledge and breaking silos" [P3].

In one quantitative study [P6], a statistically significant difference was detected when comparing with a non-Agile large-scale project: "There was, however, a statistically significant difference between the two projects in the extent Agile practices contribute to knowledge sharing" [P6].

In another case, by implementing Scrum of Scrums, the routine "has forced increased communication" [P12]. In the Norwegian government case, two routines increased collaboration and coordination:

> [CoP] champion roles implied better inter-team coordination and standardization of working. [Mid sprint] demos both made ongoing communication and collaboration with the customer smooth [P10].

At the other Norwegian organization, a *Chief Product Owner* role was appointed, named Tech Liaison, which was "particularly valuable for coordinating teams" [P10]. In the same case, common demos were introduced, which "facilitated coordination because it provided an arena for creating common expectations and understanding" [P10].

In the quantitative study of Finnish organizations adopting SAFe, cooperation was the second most reported benefit [P20]. This was also reported in both cases studied where the Spotify model was used [P24, P26].

2.5.2 Overview and transparency

On several occasions, a better overview as well as transparency was expressed, ranging from an overview of dependencies to grasping the whole project. A broad awareness [P1] and a better view of the entire project was reported from the cases, stemming from, as one author described:

> …shorter development cycles and the constant delivery of value are believed to have facilitated for the project members to grasp the status of the project. [P6]

Higher transparency and visibility [P5, P6] were confirmed in two surveys: "Our results thereby confirm … [that] project visibility is increased by several Agile practices" [P5]. "The members of Project A don't consider themselves aware of the statuses of the other project teams, while the members of Project B do" [P6].

One example of a better overview was to visualize dependencies [P1]: "…all team representatives going up to the whiteboard simultaneously and creating two diagrams representing the dependencies between teams" [P1]. At BBC, cross-team product demos enabled them to "show the bigger picture very easily and helps each team gain an understanding of recent changes and new features" [P7].

At Ericsson, Sekitoleko et al. [P16] describe that "dependencies do not become more or less; they just become more obvious." [P16].

In the quantitative study of Finnish organizations adopting SAFe, the most reported benefit was "transparency" [P20], reported by 24 respondents out of 47. From the paper investigating SimCorp, the authors acknowledge the transparency by addressing that the SAFe implementation "provides a lot of visibility…about the software development". This beneficial transparency was also acknowledged at SAP, where the authors stated that "the transparency of the development processes and the broad acceptance of agile methods gained make the change a worthwhile effort" [P25].

2.5.3 Speed, focus, and efficiency

Different aspects of increased speed and focus were presented in the studies, such as shorter lead times from plans to production as well as shorter response time on error reports. At Nokia, [P5], an important benefit in the survey study due to large-scale Agile practices was "Frequent delivery/speed/responsiveness to change" [P5] as well as productivity and efficiency [P5]. One reported benefit which increased speed was reported less overhead presented in a quantitative study: "Project B has achieved a higher … effectiveness than Project A with less overhead" [P6].

Decreased time for error reports to remain open and shorter lead time for features going from plans to implementation were reported examples of better responsiveness: "the average time that external Trouble Reports remain open shows a decrease of 31%" [P11] and: "The results also show roughly a 24% decrease in the customer service request turnaround time going from the old WoW (Way of Working) to the new WoW" [P11].

In the same case, less money was also being spent per feature [P11] as a result of increased speed. In another study at Nokia, people experienced faster feedback on problems and faster help by introducing the Scrum of Scrums routine [P13].

Increased speed was also the fourth most reported benefit in the quantitative study of Finnish organizations adopting SAFe [P20].

2.5.4 Improved quality

This theme shows improved quality after implementing inter-team coordination routines. Olszewska [P11] reported an increase in trouble reports by 188 percent, which would make you think that quality decreased rather than increased. However, this number relates to trouble reports in development and not in production. It is, thus, rather evidence of more focused work on improving quality, since, if handled accordingly, fewer defects will find its way to the production environment. At BBC [P7], the choice of appointing one Business Analyst as Product Owner per team created an increased sense of focus for the teams. Also, the BBC changed the way of performing demos: "We now demo the product from an end-user perspective; this gives a better feel and flow to the whole event" [P7].

In another organization, new ways of conducting Scrum of Scrums were initiated. Based on the people working on the same feature together, temporary *Feature Scrum of Scrums* [P14] was conducted daily until the feature was implemented, which created better focus: "people with common interests and goals could share, discuss, and even solve problems together". [P14]

At a Norwegian government agency, temporary task forces were put together to solve common platform problems, which improved quality by "building a more common understanding across teams … regarding software architecture" [P15]. The book chapter about scaling up at Ericsson [P17] describes how teamwork consequently was increasingly focused on product quality.

2.5.5 Employee motivation

At a software vendor for the oil business, the large-scale way of working improved motivation. A Scrum Master explained: "They motivate themselves (…) They get clear and quick clarification, so they don't have to wait" [P12].

When scaling up, the company Ericsson focused on more stable teams, which were perceived as more fun to work in [P17]. In one reported instance, the teams reported increased trust in each other within the team [P3], which increased motivation in the team. Findings reported from the survey performed at Nokia gave the following results: "Better

Score on 'I-know-All-Peer' and 'Trust Each Other' shows that we have improved in building a high-performance team" [P13].

At Comptel, work satisfaction declined in one of the cases, while the other showed that 70 percent of the team members where satisfied less than one year after SAFe had been implemented [P21]. Increased motivation and "improved morale" [P23] were evident at SimCorp, where SAFe was also implemented.

The Communities of Practice named "guilds", implemented at Spotify were seen as giving the employees "a strong sense of belonging and fun of being with colleagues" [P26].

2.5.6 Routines not helpful for teams

One of the drawbacks in the case organizations is that several routines are not helpful for the teams and seen as not fitting with the work process. In one case [P2], study participants felt that the discussions in the Scrum of Scrums had the wrong focus, as the author explained from their interviews:

> They pointed out that in most of the Scrum of Scrums meetings, it is difficult to have a thorough discussion and arrive at a good conclusion. Most of the time, they have to close the meetings when they get into an interesting technical discussion. [P2]

The same experience was to some extent reported from Paasivara and Lassenius [P13]. In the studied case [P13], a routine normally intended for a single team (the *demo* or *review*) was performed with all teams together in a collocated area. This was perceived as more intended for managers than for the teams:

> Each team had around 10 minutes to present results. This way of "demonstrating" was hugely criticized by the interviewed team members: they felt that a presentation did not show whether the software was good or not and that you even could cheat — tell that something worked even though problems remained. Some even felt that the common demos were arranged just for the managers to follow the progress and did not see any benefit from those to the teams. [P13]

Also, common retrospectives were not seen as useful for the teams at all:

> Huge problems remained, and common retrospectives were considered useless. Since participation was voluntary, the team members voted with their feet, and finally the meeting had only very few participants. [P13]

At Ericsson, joint meetings with stakeholders and managers were also not seen as helpful for the team since team members: "claimed they do not get the opportunity to express their burning issues or raise vital questions" [P16].

At the Norwegian Pension Fund, coordination routines that were first implemented, such as "open space", was later abandoned since, as a respondent expressed it, "in total, I feel that we did not get much out of it. Maybe we got to know each other better. Maybe." [P18].

In the quantitative study of Finnish organizations, one of the biggest obstacles was that "the SAFe model has not been fitted to the organization" [P20]. The same problem was also put forth at the SAFe implementation at Comptel [P21].

2.5.7 Conflicts between teams

Some studies presented an increased level of conflicts between teams. There were several reasons for this: Babinet [P1] stated that a conflict occurred when one team made a change in the software: "A common example is when one team does architectural refactoring that requires supporting changes by other teams" [P1].

Dependencies also led to conflicts between teams [P1]. The same problem was reported from Ericsson: "too many people involved in the same code … can end up as conflicts in other teams" [P16].

From one case organization, Evbota et al. [P2] reported a conflict due to changing members within teams:

> …removing team members from an established team destabilize its spirit, which has been built over the period of close teamwork. In addition to that, the team spirit in the receiving team can also be impacted. [P2]

A problem related to differences in individual team members' competences was seen in another case: "when a member of the team is continuously consulted for his or her knowledge by many other teams" [P8].

Another side of this problem was pinpointed in the Norwegian government organization. Although ad hoc meetings solved problems, "a negative aspect is that it was dependent on each individual" [P10].

Also, if the Product Owner was split to be responsible for two teams, it could cause conflicts as reported by Lowery [P7]: "Product Owners gave contradictory directions, often on a daily basis". [P7].

From the survey study by Laanti et al. (2011), the third most highly perceived challenge was: "Dependencies, cooperation…" [P5]. An example of this reported from Siemens HS was that teams would:

> …take on additional goals while other teams would fall short of their goals. We had envisioned that the team leaders would seek opportunities to help other teams within the program before pushing ahead independently. [P9].

2.5.8 Stress

Notions of increasing stress were reported in some of the papers. In one organization, backlogs were kept both in the teams and on an overall level, causing stress in the teams: "potential inconsistencies between different backlogs as well as lack of their transparency are a big challenge" [P2].

In another case organization, team members felt moderately stressed, which was caused especially by demos in the project called B, which was a large-scale ASD project:

> The average levels of negative stress and pressure stated by the survey respondents from Project B are all slightly higher than the average levels stated by members from Project A [P6].

However, negative stress caused by pressure to report progress in daily meetings, as reported by Stray et al. (2012), was not present to a large extent. Paasivara et al. [P14], on the other hand, reported that the implementation of Scrum of Scrums caused frustration regarding the outcomes of these meetings: "in general you don't find solutions from there anyway" [P14].

At Ericsson, the stress and frustration were due to expectations about team members' competences: "expectations on deep technical competences and broad craftsmanship were sometimes seen as almost overwhelming" [P17].

Another frustrating experience was that there was less of "try and do something new", according to Wigander [P17]. For Product Owners, the increased stress was due to tension around their availability for other roles in the business unit [P4]. In another paper from another part of Ericsson R&D, Paasivaara et al. [P22] reported that the "journey towards agile had been challenging and stressful for everyone as a vast number of changes had been implemented". This was also described at SAP, where "programmers complained about the pressure both from management and from within their teams" [P25].

2.5.9 Limited autonomy

In the beginning of the scale-up process at BBC, managers relied on a single Product Owner for several teams. The impact of this was that: "he [the Product Owner] never actively empowered the teams to make their own decisions" [P7]. This was also evident at Ericsson, where a reported perceived drawback when scaling up was lesser flexibility for the teams [P17].

Limited autonomy is not only a problem with the development team, but also for other roles. The Product Owner role could be problematic in large-scale Agile settings, as described in three case organizations observed by Hobbs and Petit [P4] regarding mandate: "most Product Owners need to address questions to the customer organization" [P4].

The autonomy for the Product Owner is sometimes further limited by the introduction of a *Chief Product Owner* or *Product Manager* role, according to 60 percent of the survey answers in [P4].

The open work area at the Norwegian Pension Fund allowed for faster coordination, but "the extra roles (such as functional architect) and arenas (such as the Open Space) led to less autonomy for the team" [P18].

At Comptel, where SAFe was implemented, team members also perceived limited autonomy "as they could no longer decide some things on their own, such as the sprint length" [P21].

2.5.10 Unclear responsibilities

In many cases where ASD is scaled up, new roles are added for coordination purposes (Gustavsson, 2017). This can sometimes lead to unclear responsibilities, as in the case of a Healthcare software vendor [P19]. In lessons learned from this case, the authors recommended: "If you have product managers and project managers … then try to define clear roles, responsibilities and focus areas" [P19].

This issue was also evident in the survey study of Finnish organizations implementing SAFe where complaints about leadership and change of leaders was the fourth most reported problem [P20].

In detailed descriptions from the SAFe implementation at SimCorp, former managers had to apply for the newly implemented SAFe roles, and two years later: "some individuals were still challenged with this change of role" [P23].

The investigated LeSS implementation at a German automotive manufacturer showed that two of the main reported challenges were "Missing

transparency regarding roles and responsibilities" and "Dual leadership of the PO role" [P27].

2.5.11 More difficult to prioritize

There were several reasons for problems regarding prioritizations. Martini et al. [P8] describe problems with prioritizing when teams have several customers:

> …the team has to follow requirements defined by a reference architecture, and it has to negotiate various kinds of requirements with other teams, i.e., the team has more than one customer: product management, architects, and the other teams. [P8].

At BBC [P7], two problems occurred which had an impact on prioritization:

> Uncoordinated teams … often deliver what they want, not what the overall business actually needs. Product Owners were retained to act as product specialists more than pure Product Owners who [normally] set priorities. [P7]

At Ericsson, teams were not willing to express time estimations of deliveries, claiming it to be impossible and a waste of time. The large-scale setting led to significant amounts of troubleshooting, which made it hard to anticipate available resources. This led teams to become skeptical about estimations in general, and the teams less willing to do estimations at all:

> …the amount of content (i.e., product backlog items) is too big. This experience has lead teams to become skeptical about estimation in general, and some choosing not to do any estimates at all. [P2]

Managing requirements at SAP was transformed into a three-level hierarchy for teams to manage. This did not work as expected but instead proved to make it more difficult to prioritize [P25].

2.5.12 Selfishness

Sekitoleko et al. [P16] reported a drawback due to the amount of time spent on inter-team coordination routines which could be summarized as signs of *selfishness*: "Some team members do not want to share knowledge because they fear that others will start seeking help from them more often" [P16].

Paasivaara and Lassenius [P13] observed Scrum of Scrums where employees just stated "nothing to report", even if this might not be true just because it was easier for them. When many teams met in the Scrum of

Scrums, many participants considered it a waste of time and spoke as little as possible to be able to get back to work quicker:

> Most of the time everybody goes [to Scrum of Scrums] and just says they've got no problems ... they just took too long, because we've got so many teams here ... it seems to me like people don't want to talk so much at these things. [P13]

Almost the same comment was made in another paper by Paasivara et al. [P14] where "many teams report 'No problems', which is not always entirely true, but the result of assuming that the other teams do not have to know" [P14].

2.5.13 Too much time spent in meetings

Additional routines for inter-team coordination means more time spent in meetings for many employees (Gustavsson, 2018).

In the SAFe implementation at Comptel, respondents confirmed that many team members saw these added routines "more like an overhead to them" [P21].

This problem was also confirmed in the implementation of LeSS at a German automotive manufacturer. Especially, the authors presented the problem of Product Owners being held up in too many meetings. One Scrum Master described the problem as "Many people are simply arrested in thousands of other meetings and if the PO is missing at a meeting such as sprint planning, this is quite sub-optimal" [P27].

2.5.14 Shyness

In one of the papers describing scale-up at Ericsson [P16], another drawback was presented: "team members are too shy to either ask or provide information during meetings, thus causing important information to be ignored" [P16].

This is confirmed by Wigander [P17], who expressed this in another way reporting from lessons learned at Ericsson: "There … [is] a need for higher social skills than before" [P17].

2.5.15 Clashes with surrounding organizations

One of the challenges expressed in the study at Ericsson R&D was to have "Agile teams in a waterfall organization" [P22]. One team member expressed this drawback for the team, where "we have different checkpoints [compared to the surrounding organization] and meetings before we can move on" [P22].

A similar drawback was reported as one of the three major challenges reported from the SAFe implementation at SimCorp: "interfaces to other parts of the SimCorp organization" [P23]. The authors described the problem of not reaching full potential when supporting functions did not work in the same way, using other routines and another pace of work.

2.5.16 Short summary

Most of the articles on large-scale Agile ways of working focus on the transformation process or on adapting and tailoring the Agile ways of working, not on actual impacts. Few impacts were reported, whether on an organizational level or for teams. Only 27 papers were identified that presented any impacts on teams due to implementing large-scale Agile ways of working in the organization. None of these papers focused directly on impacts on the teams. Still, they provided information on impacts in several ways, but mostly on the organization as a whole and not on individual teams.

Consultants claim several benefits of Agile ways of working expressed as measured numbers, such as productivity increases of 20 to 50 percent, decreased time to market with 30 to 75 percent, 10 to 50 percent more motivated employees, and 25 to 75 percent defect reduction (Scaled Agile, 2020). However, none of these claims have been tested by scientists. Although several comments have been put forth regarding "increased speed", no research on measuring speed or productivity has been found. However, one could argue that the reported decrease in trouble reports (Olszewska et al., 2016) might lead to increased speed. Further research needs to confirm if that is the case, however.

Regarding drawbacks, none of the possible drawbacks are mentioned by consultants who suggest frameworks to be implemented in large-scale settings.

3 Coordination theory

There is a large body of existing theory in organization studies, management science and IS research that could be applied to better understand and explain different aspects of ASD (Strode, Huff, Hope, & Link, 2012). Research on ASD has matured in the past years, but several questions and concepts are still not fully understood (Ågerfalk et al., 2009). Coordination in large-scale ASD is one such concept where coordination theory could help to develop our understanding further (Strode et al., 2012; Moe et al., 2016). In this chapter, I discuss the development of coordination theory, an area with roots as old as organizational management. The theory is helpful to explain concepts and routines for coordination, albeit its lack of strength for prediction. Coordination theory will help to explain *how* inter-team coordination is performed and tailored. This section presents the first part of the theoretical framework that will guide the study of how and why organizations are performing inter-team coordination routines in large-scale ASD.

Coordination problems are caused by interdependencies between various elements of a situation that constrain how particular tasks are performed. An early definition of coordination can be found in Van de Ven et al. (1976), who define coordination as "integrating or linking together different parts of an organization to accomplish a collective set of tasks". This definition is rather wide since it does not specify what *parts* really means. To further specify the concept, Malone and Crowston (1994) suggested that the *parts* rather should be seen as activities that are constrained by dependencies. Their (widely used) definition of coordination is therefore "managing dependencies between activities" (Malone & Crowston, 1994, p. 90). One could discuss whether coordination only entails coordination of activities or whether this definition is too narrow. For example, as Dingsøyr, Moe, and Seim (2018) put forth, coordination of knowledge is also important for the Agile organization. However, I argue that a definition of coordination should include both coordination of dependent activities, as proposed by Malone and Crowston (1994) as well as knowledge. Hence, by interpreting *parts* as both dependent activities and knowledge, the definition by Van de Ven et al. (1976) becomes clear and useful for this study.

The dependency problems with activities entail *actors* with *interdependent tasks* to perform coordination, sometimes requiring or even creating *resources* to overcome the constraints of these dependencies (Crowston, Ru-

bleske, & Howison, 2003). Dependency can be seen as the inverse of power, since the power of one actor to decide autonomously what and how to perform an action becomes limited.

Thompson (1967) categorizes dependency in terms of workflow between actors. He describes three kinds of dependencies:

1) *Pooled interdependence*, where the organization draws resources from a shared source. Typically, a problem in organizations where a few experts have most of the knowledge or where many actors depend on a few resources.
2) *Sequential interdependence*, where the output of one team is input to another. People in the latter part of the chain become highly dependent on previous work processes, such as a marketing department that needs to have finished implementations of software solutions.
3) *Reciprocal interdependence*, where one team's output depends on another team's input, *and* vice versa. An example is a hotel where the reception is dependent on information from the cleaning team to check in new guests, while the cleaning team needs information on what rooms to clean.

Solutions to coordinate work has been an important research topic in organizational and management studies for a long time. March and Simon (1958) divided the solutions into three areas (or levels): Coordination by 1) standardization, 2) planning, and 3) feedback or *mutual adjustment* as it was rephrased by Thompson (1967). March and Simon (1958) explain that the need for communication increases for every level of coordination solution so, since communication takes time and resources, the most efficient way to solve coordination issues is to focus mainly on standardization. According to Thompson (1967), these ways to coordinate map how to solve the above-mentioned dependency problems, i. e. pooled interdependence problems should be solved by standardization, sequential interdependence by plans, and reciprocal interdependence by mutual adjustments.

Thompson (1967) also made a distinction between horizontal and vertical coordination. The horizontal coordination is between employees on the lowest level of the organizational hierarchy, and, according to the author, the key to this area of coordination is mutual adjustments. Vertical coordination is managers or management teams deciding from the top down. All coordination issues that cannot be solved by mutual adjustments should be dealt with by the management team (Thompson (1967).

These earlier writings do not consider teams, but rather coordination on a personal employee level. Van de Ven et al. (1976) added the concept of *team mutual adjustments* and presented two interesting claims based on a survey study: the amount of coordination effort depends on task uncertainty, not the number of people involved. Also, as the number of people and teams increase in an environment of high task uncertainty, there should be a lesser need for vertical coordination and more need for *rules*, *policy*, and *horizontal coordination*, i. e. scheduled and unscheduled meetings to resolve dependencies. Once implemented, the impersonal coordination (rules and policies) are codified and require minimal verbal communication between people (Van de Ven et al., 1976). Examples include pre-established plans, process documentation, intranet pages, and roadmaps. Coordination by mutual adjustment or feedback, on the other hand, should be based on informal communication. In the personal mode, individual role occupants serve as the mechanism for making mutual adjustments through either vertical or horizontal channels of communication. The mechanisms for vertical communication are usually line managers.

The presented ideas from Thompson (1967), Van de Ven et al. (1976), and other authors such as Mintzberg (1983) could be seen as the basis for forming an interdisciplinary Coordination Theory which was presented by Malone and Crowston (1994). Malone and Crowston (1994) analyzed group action in terms of *actors* performing *interdependent tasks*. These tasks might require or create *resources* of various types. In this view, actors in organizations face *coordination problems* arising from dependencies that constrain how tasks can be performed. To overcome these coordination problems, actors must perform additional work, which Malone and Crowston (1994) called *coordination mechanisms*. There are often several coordination mechanisms that can be used to manage dependencies. In developing coordination theory, Malone and Crowston (1994) describe coordination mechanisms as relying on other necessary group functions, such as decision making, communications, and development of shared understandings and collective sensemaking.

The main claim of coordination theory is that dependencies and the mechanisms for managing them are general. In other words, a given dependency and a mechanism to manage it will be found in a variety of organizational settings. For example, a common coordination problem is that certain tasks require specialized skills, thus constraining which actors can work on them. This dependency between a task and an actor arises in some form in nearly every organization. Coordination theory suggests

identifying and studying such common dependencies and their related coordination mechanisms across a wide variety of organizational settings. Given an organization performing some task, one way to generate alternative processes is to first identify the particular dependencies and coordination problems faced by that organization and then consider what alternative coordination mechanisms could be used to manage them (Crowston et al., 2003). Etcheverry, Lopistéguy, and Dagorret (2001a; 2001b) suggest that it is helpful to organize mechanisms into coordination *patterns*. They define a pattern as a solution to a problem in a given context, which suggests that coordination mechanisms could rather be organized by coordination contexts, not only dependencies per se. Etcheverry et al. (2001a; 2001b) call out for further work to bring together and organize the mechanisms into context-based patterns.

Malone and Crowston (1994) conceptualized dependencies as arising from shared use of resources by multiple tasks, thus providing a conceptual basis for a typology of dependencies. One contribution of coordination theory is a typology of dependencies, and sub-dependencies, and related coordination mechanisms:

- *Task-resource dependencies*. Most tasks require resources, if only the effort of the actor performing the task, and an important class of coordination processes manage the assignment of resources to tasks.
- *Producer-consumer dependencies*. Another major type of dependency is a producer-consumer dependency, where the output of one task is the input to another. Malone and Crowston (1994) note that such dependencies often impose additional constraints. Three sub-dependencies are put forth as *usability*, ensuring that the output is of a form usable by the next task, and *transfer*, ensuring that the output is available to the consumer when needed. The third is *precedence*, which means that the actor performing the second task must learn when the resource is available, and the task can be started.
- *Task-subtask dependencies*. Third, some tasks are decomposable into subtasks. A key issue in any decomposition, however, is ensuring that the overall goal is satisfied by the performance of the subtasks.

One could discuss whether the coordination theory presented by Malone and Crowston (1994) really is a theory. Researchers who argue that it is *not* a theory assume a definition of theory similar to the one stated in the *American Heritage Dictionary of the English Language*:

> A set of statements or principles devised to explain a group of facts or phenomena, especially one that has been repeatedly tested or is widely accepted and can be used to make predictions about natural phenomena. (Houghton, 2000, p. 441)

But the critical voices overlook the fact that theories include scientific concepts as well as possibilities for predictions (Kaplan, 1964). Theory, in this sense, helps make sense of data and makes observed events meaningful. Concept and theory generation go hand in hand, and an initial set of concepts guides the search for data and possibilities for predictions. Crowston et al. (2003) argue that coordination theory definitely includes a set of concepts that can be used to label phenomena and link them to others, and thus fulfills the important function of theory. In its current state, though, Crowston et al. (2003) would describe coordination theory as a pattern model (Kaplan, 1964), meaning that it seeks to explain phenomena by showing how they fit into a known pattern.

3.1 Planned coordination versus emerging dependencies

Neither March and Simon (1958), Thompson (1967), Van de Ven et al. (1976), nor Mintzberg (1983, 1998) describe any routines or process solutions for how to accomplish the important area of mutual adjustments, only that it is critical for horizontal coordination. The criticism against this earlier work (e. g. Jarzabkowski et al., 2012 and Okhuysen & Bechky, 2009) is that the focus is on the *static view* of coordination where detailed solutions to *emerging* coordination issues are mostly overlooked. Also, Jarzabkowski et al. (2012) point out that there is not much research on how these coordination routines change over time.

As an example, practices for mutual adjustments are not as detailed as areas of standardization where Mintzberg (1983, 1998) made a distinction of four different solutions: standardization of work processes, outputs, skills, and, finally, norms (added in 1998). Practices for dealing with emerging, unplanned contingencies have collectively been labeled *mutual adjustments* (Thompson, 1967), *lateral relations* (Galbraith, 1973), or *ad-hoc coordination* (Donaldson, 2001). These practices have been difficult for scholars to measure and therefore remained largely unexamined (Okhuysen & Bechky, 2009). This is unfortunate because unplanned contingencies and responses to them are important for the efficiency of workflows and processes in organizations (Okhuysen & Bechky, 2009). Therefore, Okhuysen and Bechky created a framework of five areas of

coordination to manage emerging dependency issues. The framework can be used to analyze and categorize coordination for dealing with emerging dependency issues into the following areas: Plans and rules, Objects and representations, Roles, Routines, Proximity (location). This is the framework I used in this thesis to explain the various forms of coordination prescribed by SAFe (see 2.3).

3.2 Inter-team coordination

According to Hoegl et al. (2004), there are mainly two forces that create coordination needs between teams in large software development projects: (1) task interdependencies and (2) changes occurring during the development process.

Task interdependencies refer to the direction of a workflow relationship between two teams. As a team depends on input from another team to accomplishing its own task, the work in one team has implications for the work and progress of other teams. While some coordination needs are possible to plan in advance, software development is always characterized by changes (Hoegl et al., 2004), which often affect the work of several teams. The complexity and uncertainty of development processes, based on interdependencies between teams and frequent changes, can only be dealt with if information is exchanged between teams.

Hoegl et al. (2004) showed that although coordination with other teams might take time and resources, it has, to some degree, a positive impact on team performance. Hoegl et al. (2004) measured team performance as a combination of schedule performance (delivering on time), adherence to budget, and quality. Inter-team coordination showed a significant positive relationship to schedule performance, but not to quality, and a negative relationship to adherence to budget.

3.3 Synthesis of coordination theory

As presented in this chapter, concepts from coordination theories are helpful in theorizing about routines and tailoring, especially regarding performative aspects of routines (*how* they are performed). Coordination theory will help to explain how inter-team coordination is performed and tailored in the different cases studied in this thesis.

To sum up, coordination theory, as presented by Malone and Crowston (1994), is an analytical, typological framework that can be used to analyze and categorize dependencies among resources and tasks in an or-

ganization. It also serves as a framework to analyze the implemented coordination mechanisms and how they are conducted, i.e., the routines. These analyses can further be used to suggest alternative coordination, created by substituting one coordination mechanism for another. The dependency concepts in coordination theory from Malone and Crowston (1994) and Van de Ven et al. (1976) will be used to analyze the different sorts of dependencies in this study.

The focus of earlier research into coordination has been the *static view* of coordination, where detailed solutions to *emerging* coordination issues have been mostly overlooked. Okhuysen and Bechky (2009) therefore suggested an extension to coordination theory through an analytical framework of five areas to be used to identify coordination for emerging dependency issues. In this study, inter-team coordination routines are also analyzed based on whether they solve planned dependencies or emerging dependency issues.

In the next chapter, I will discuss new institutional theory, which is used for conceptualizing reasoning and decision making related to coordination routines on an inter-team level (*why* they are implemented and tailored).

4 New institutional theory

Finding and choosing appropriate theories for a study could be somewhat of a roller coaster ride. As described in the previous chapter, I chose theories on coordination to theorize on performative aspects of routines. Unfortunately, these theories are not enough to explain why inter-team coordination routines are implemented in different ways, although they are trying to solve the same kind of problems. The theories explain that different routines can be used for the same purpose (e.g., Malone & Crowston, 1994) but do not describe why these choices are different. Regarding reasoning and decision-making, I first turned to theories on the dynamics of organizational routines (Pentland et al., 2012). As described earlier, Feldman and Pentland (2003) helped me in defining the concept of organizational routines. Pentland et al. (2012) also made me understand the importance of differences in the ostensive aspects of implementing routines (*why* they are implemented and changed). The ostensive aspects of implementing routines provides names and concepts, but in order to understand the underlying reasons for these differences, I had to search for other theories. In this search, I found papers on institutional logics (e.g., Thorén, Ågerfalk, & Edenius, 2014) and realized that this was what I was looking for.

Therefore, institutional logics concepts stemming from new institutional theory (Thornton & Ocasio, 2008) will be presented in this chapter as the second part of the theoretical framework that will guide this study. Rather than a coherent theory, the new institutional theory is a loosely coupled body of knowledge accumulated in several streams of research, for example highlighted in Doležel (2018) and Thornton & Ocasio (2008). Applying new institutional theory offers several advantages for IS researchers (Rowlands, 2008). For example, it offers a "structural and systematic understanding of how technologies are embedded in complex interdependent social, economic, and political networks, and how they are shaped by such broader institutional influences" (Orlikowski & Barley, 2001, p. 154).

Institutional logics is a concept in sociological theory and organizational studies that focus on how broader belief systems shape the cognition and behavior of actors. Friedland and Alford (1991), who introduced the concept, identified several key *institutions* such as the nuclear family and bureaucratic state, each guided by a distinct *institutional logic*. The main innovation of Friedland and Alford was to conceptualize society as an

interinstitutional system (Thornton & Ocasio, 2008), meaning that we all need to navigate between logics prevailing from different institutions present around us.

In the next section, I will define what an institution is according to new institutional theory. Thereafter, I will discuss institutional orders, institutional pluralism, and finally present a synthesis of new institutional theory.

4.1 Institutions in new institutional theory

Jepperson (1991) defines an institution as "an organized, established, procedure" (p. 143). Institutions are not physical realities driving human action. Rather, institutions are enacted through reproduced patterns of activities by individuals. North (1990) emphasized how institutions can force our behavior by describing them as "humanly devised constraints that shape human interaction". A set of goals, values, and prescriptions associated with a specific institution form a rationale or institutional logic (Friedland & Alford, 1991). Therefore, to be able to understand both individual and organizational behavior, it must be located in an institutional context which both regularizes behavior and, at the same time, provides an opportunity for agency and change (Thornton & Ocasio, 2008).

The institutional logics perspective is commonly used in IS research, but what constitutes the studied *institutions* is not always clearly described. As Thornton and Ocasio (2008) express it, "what constitutes an institution remains an unresolved conceptual issue for the field" (p. 121).

In a recent paper named *Institutions: Everywhere But Not Everything* (Ocasio & Gai, 2020), the authors acknowledged the problem addressed in the title, despite the different attempts to define institutions. However, an updated attempt to clarify what defines an institution was presented by Ocasio, Thornton and Lounsbury (2017) as a "taken-for-granted, normatively sanctioned set of roles and interaction orders for collective action" (p. 526). This definition implies that institutions have a collective agency and are self-reproducing.

4.1.1 What institutions are not

Since an institution could be referred to as an actual physical entity, such as nation-states or universities (Ocasio & Gai, 2020), the use of the concept in organizational research could be confusing.

First, institutions are not the same as organizations (Doležel, 2018). As described previously, institutions shape human interaction (North,

1990). Organizations, on the other hand, are formalized entities officially represented by their administrators. Administrators direct employees towards a goal, such as to provide service in a government agency or profit generation in a business organization. For example, journalism is an institution, whereas a newspaper is an organization (Ferrer Conill, 2018).

One might propose that institutional logics would be the same thing as organizational culture. There is definitely some overlap between the two concepts (Doležel, 2018). Both concepts focus on goals, values, and behavior in organizations (Thornton & Ocasio, 2008). The difference, however, is that institutional logics is focused on behavior in many organizations, not one. To use the same example as above, an institutional logics perspective focuses on the logics of journalism, not the logics of a specific newspaper (Ferrer Conill, 2018).

4.2 Institutional orders and logics

Thornton, Ocasio, and Lounsbury (2012) revised and expanded the research on institutions into defining seven ideal types of *institutional orders* to propose an interinstitutional system. The seven institutional order ideal types are: the family, the community, the religion, the state, the market, the profession, and the corporation. These institutional orders represent the governance systems by which actors in the institution form their sense-making choices. Ideal types of institutional order do not describe an organizational field but are theoretical models for comparing various meanings in a location with a definable boundary (Thornton et al., 2012). They are deliberately simplified to be used for comparative analysis of phenomena not restricted by the events of the selected cases (Thornton & Ocasio, 2008).

Each institutional order is composed of *elemental categories* (Thornton et al., 2012) or *dimensions* (Berente & Yoo, 2012), specifying organizing principles that shape preferences and interests. Examples of these dimensions are: root metaphor, sources of legitimacy, authority, and identity, as well as the basis of attention and strategy (Thornton et al., 2012). To clarify, Table 3 presents these six dimensions for two of the ideal types of institutional orders (the community and the profession) as described in Thornton et al. (2012).

Table 3. Institutional order ideal types for community and profession

Dimension	The community	The profession
Root Metaphor	Common boundary	Profession as relational network
Sources of Legitimacy	Unity of will. Belief in trust & reciprocity	Personal expertise
Sources of Authority	Commitment to community values & ideology	Professional association
Sources of Identity	Emotional connection. Ego-satisfaction and reputation	Association with quality of craft. Personal reputation
Basis of Attention	Personal investment in group	Status in profession
Basis of Strategy	Increase status & honor of members & practices	Increase personal reputation.

Different authors have used different sets of dimensions to describe ideal types. Berente and Yoo (2012) used four dimensions while Thornton and Ocasio (2008) used twelve dimensions. I chose to present ideal types using these six dimensions because I believe they are sufficient to present the ideal type clearly enough while they also do not overlap.

Revising the original term *institutions* into a theoretically abstract and analytically distinct set of *institutional orders* makes it useful for studying multiple logics in different parts of society as well as within organizations. Besides defining institutional orders, Thornton et al. (2012) developed the definition of institutional logics into "the socially constructed, historical patterns of cultural symbols and material practices, including assumptions, values, and beliefs, by which individuals and organizations provide meaning to their daily activity, organize time and space, and reproduce their lives and experiences" (p. 2).

4.3 Institutional pluralism

While much earlier work focused on ambiguity caused by conflicting institutional logics (Friedland & Alford, 1991), recent research displays a more pluralistic approach by focusing on several competing logics, a situation known as *institutional pluralism*. The co-existence of multiple logics also resides within a single organization. By studying how organizations are composed of multiple logics and multiple forms of institutionally-based rationality, researchers can provide new insights into the variation and tailoring of routines (Thornton et al., 2012). The institutional logics perspective offers valuable insights into important intra-organizational routines, to understand institutional complexity due to conflicting or incon-

sistent logics within organizations. It offers a lens for analyzing goals, values, and prescriptions that underlie and legitimize behaviors of teams and individuals (Berente & Yoo, 2012).

The institutional logics perspective is, therefore, useful in my study as an analytical tool to explain how actors are influenced by competing logics in an interinstitutional system (Thornton et al., 2012). Actors do not simply apply institutional rules. They navigate institutional orders in their everyday routines in ways that are consistent with individual rationality based on different logics (Berente & Yoo, 2012).

4.4 Synthesis of new institutional theory

To summarize, an institution is not a physical entity but organized, established procedures enacted through reproduced patterns of activities by individuals (Jepperson, 1991). A set of goals, values, and prescriptions associated with a specific institution form a rationale or institutional logic (Friedland & Alford, 1991). Thornton et al. (2012) further developed the research on institutions into defining seven ideal types of institutional orders to propose an interinstitutional system. The institutional orders represent the governance systems by which actors in the institution form their sensemaking choices. Each institutional order is composed of dimensions (or elemental categories) specifying organizing principles that shape preferences and interests. By studying how organizations are composed of multiple logics, and multiple forms of institutionally-based rationality, researchers can provide new insights into the variation and tailoring of routines (Thornton et al. 2012).

As presented in this chapter, the concepts from new institutional theories are helpful in theorizing about routines. Therefore, in order to unfold a deeper understanding of *why* inter-team coordination routines are tailored and performed in various ways, the lens of institutional logics is applied in this study. The concepts of new institutional theory described in this chapter are used in this study in two ways. First, I put forth a theoretical argument proposing institutions and institutional logics to be used for analysis in large-scale ASD (chapter 6). Second, the theoretical propositions are used for analysis of the cases studied (chapter 7, 8, 9, and 10).

5 Research Methods

In this chapter, the methods employed in this thesis are discussed. To know what methodologies are appropriate, one needs to understand the benefits of different methods and how they can help to answer one's research questions. Therefore, I have had to use different methods for different aspects of my studies.

Studying complex empirical realities is difficult (Davies, Manning, & Söderlund, 2018). This has created a growing need for cross-fertilization and collaboration between research disciplines (Davies et al. 2018). I strongly believe that sharing of concepts and empirical domains with other disciplines will promote the development of our scientific knowledge and, as put forth by Davies et al. (2018), the problem of *ideological distancing* (or *self-referential ideologies*, citing scholars in the same field) may, more or less intentionally, exclude other relevant perspectives and interpretations.

This study focuses on the phenomenon of inter-team coordination in large-scale ASD within the research field of Information Systems. The unit of analysis is the organizational routines (Pentland & Feldman, 2005) implemented for carrying out this coordination between teams. To be precise, the research field could be expressed as information systems development and the project management of systems development. Therefore, I have intentionally tried to cross-fertilize concepts and research from Software Engineering, Project Management, and Organizational studies with IS research. By involving and embracing both research from these other disciplines as well as attending conferences and discussions with other researchers, I have tried to avoid the risk of *encapsulation* (Davies et al. 2018), i.e., failing to recognize contributions from neighboring disciplines within the same research area.

In the first section of this chapter, the chosen research approach is motivated and discussed. To gather and synthesize previous research on large-scale ASD, I performed the steps to conduct a systematic literature review (SLR). Therefore, I explain the methodological steps to perform an SLR in the second section of this chapter. The results of the SLR are presented above in chapter 2.5 (Impacts on teams in large-scale ASD). In the third part of this chapter, the case study approach employed in this thesis is presented and discussed. Following that, the data collection is described, the analytic strategy presented, and finally, rigor and research ethics are discussed.

5.1 Epistemological and ontological considerations

Studying critical realism has helped me get a sense of how to produce knowledge from my research. Critical realism is a metatheory, or worldview, of knowledge (epistemology) and understanding of the nature of reality (ontology) originally presented by Bhaskar (1975). Critical realists believe that there are unobservable events that cause the observable ones, which means that the social world can be understood only if people understand the structures that generate such unobservable events. There is a reality that exists independent of its human conception, but this reality is transcendental in the sense that there are aspects to reality that are not readily apparent.

This view allows the scientist to distinguish between the event and what causes it. According to critical realists, an individual creates the conditions necessary for the observable event, but the results are caused by underlying mechanisms (unobservable events). To put it in another way, "phenomena possess latent powers due to their structure; they are effected through some sort of mechanism, the law by which the structure exercises its powers. When these powers are exercised, they have effects in the world" (Okoli, 2015, p. 25).

In social science, culture and society are generated by human activities. Society is continuously changing due to the dynamic nature of human actions, causing a mutually influential relationship where humans shape society, which in turn affects human activities. The social structures are open and cannot be artificially controlled in a laboratory setting, and, unlike natural laws, rules of culture and society are not universal but applicable only in a certain location and time. Most natural sciences can conduct experiments in closed systems, whereas the social contexts of social sciences cannot be closed in that way since they operate in open systems. In a closed system, all (or nearly all) mechanisms are accounted for, and actions are expected to be highly predictable, but that is not the case with social science where the complexity of human social interactions is studied. Therefore, the critical realism approach can best be used for its explanatory benefits, rather than for predictive purposes (Danermark et al., 2003).

Critical realism requires a deep understanding of any social situation, going beyond the observable and investigating mechanisms behind events. The theory focuses on ex-post explanations to explain underlying reasons for events to occur, such as psychological, cultural, or organiza-

tional reasons. To use a critical realist approach in research is to explain complex social events by understanding possible explanations and ruling out other potential explanations. The researcher does not seek universal social laws that apply uniformly in all situations, but instead realizes that because of the complexity of social systems, social science is highly context-specific. However, although a universal law is not searched for, the researcher looks for patterns or tendencies that regularly hold to provide a general basis for action (Danermark et al., 2003). The following sections explain how I have gone about identifying such patterns.

5.2 Research approach

In IS research, Järvinen (2008) proposed a taxonomy for research approaches, based on a previous taxonomy by March and Smith (1995), to guide junior scientists when choosing an approach based on their research question. The chosen research approach based on Järvinen's (2008) taxonomy is presented in Figure 4.

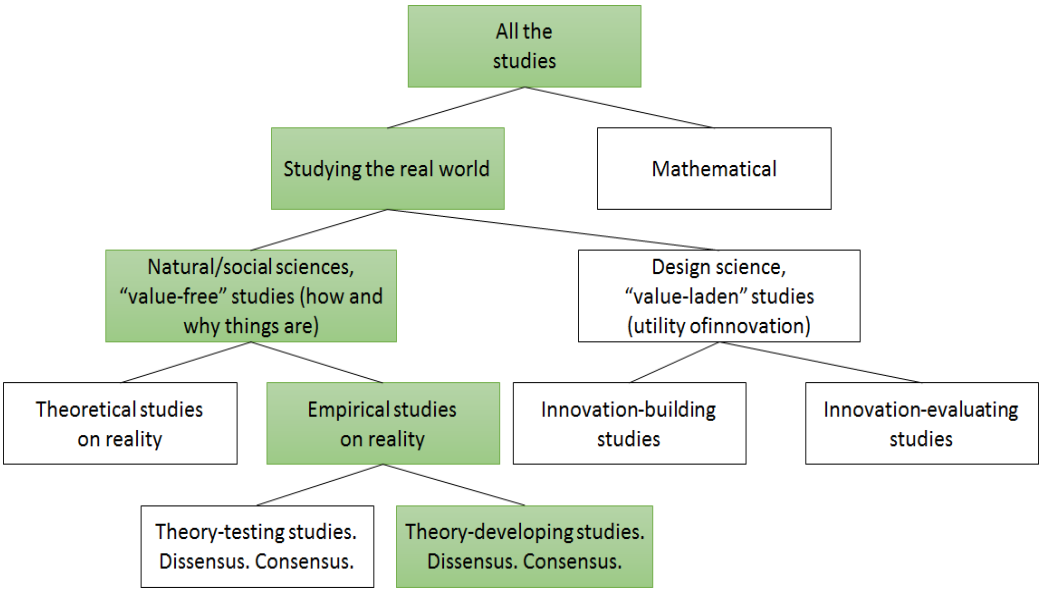

Figure 4. Research approach (green indicates this study).

The multiple-case study presented in this thesis is what Järvinen (2008) classifies as a theory-developing study, which includes multiple-case study methods, content analysis, and longitudinal study methods, to name a few. The research approach chosen for this thesis is presented by the green-colored boxes in Figure 4. Theory-developing studies are mostly

carried out through qualitative analysis of interview transcripts, artifacts such as documents or plans, and on-site observations (Järvinen, 2008).

The research presented in this thesis is based on the project-as-practice approach to research in project management research (Blomqvist et al., 2010). The approach implies a focus on micro-level practices and routines, attempting to understand how practitioners act and make sense of their situation. Project-as-practice research focuses on the actual activities performed by individuals and teams in projects, and how these align with or deviate from established norms, routines, and behavioral expectations (Manning, 2008). Focus is on the ways in which activities are enacted and modified, negotiated, and contested by and among the actors involved. The perspective encourages researchers to conceptualize norms and ways of doing things as potentially dynamic, contextual, and subject to change, working against narrow ideologies, such as striving towards defining an optimizing model.

The practice perspective on projects was originally promoted by the *Scandinavian School of Project Studies* (Engwall, 2003; Lundin & Söderholm, 1995; Sahlin-Andrewson, & Söderholm, 2002; Söderlund, 2005). The approach has been an important foundation for the current debate on temporary organizing (Bakker, DeFillippi, Schwab, & Sydow, 2016), focusing on how temporary structures and processes affect the way individuals are coordinated within and across organizations (Hällgren & Söderholm, 2011).

5.3 Systematic literature review (SLR)

A literature review can be performed and presented in varying ways. For this thesis, I used the systematic review method as described by Kitchenham (2007) to understand what benefits and drawbacks that have already been found in large-scale ASD workplaces where inter-team coordination is key. Kitchenham was not the first to develop this method. Rather, the tradition of performing systematic reviews and meta-analysis originates from health care research where it has been used for a long time. It is often used to understand the effects of medicines by comparing several studies where the same active substances have been used. The systematic review method was instead translated for use in software engineering by Kitchenham and later further adapted by Dybå and Dingsøyr (2008) and Cruzes and Dybå (2011). Kitchenham's way of performing systematic literature reviews is commonly used for software engineering as well as for

IS research. Systematic literature reviews are a means of identifying, evaluating and interpreting available research relevant to a particular research question, topic area, or phenomenon of interest. They are appropriate for summarizing existing research for identifying gaps in the existing literature, as well as for providing background for positioning new research (Kitchenham, 2007). This conducted SLR was targeted towards identifying impacts on teams in large-scale ASD settings, but it also served as a way to identify other areas of previous research on large-scale ASD useful for this thesis. The design of the research process used in this thesis is based on recommendations from National Institute for Health Research (2011), suggesting to use templates from the PROSPERO protocol (National Institute for Health Research, 2011) and the PRISMA Flow Diagram (Moher et al., 2009) to guide the process. Although intended for systematic reviews relevant to health-related topics, these templates proved to be of great help to organize and record data along the process.

The selection of studies was made in three stages. First, a search of earlier systematic literature reviews on the topic of Agile software development was conducted to find relevant databases used by other researchers (and to ascertain that no one had made a thorough review of articles specifically of interest for my study). Second, keyword database searches were made to identify potentially relevant sources and, third, abstracts and full-text articles were manually screened. The second and third stages were performed twice: a first round was conducted in early 2018, and a second round in July 2020 to find additional papers published since 2018.

After that, data extraction was conducted, following the recommended steps for thematic synthesis in software engineering by Cruzes and Dybå (2011, p. 277). The first of the recommended steps is to extract three different sorts of data:

- "Publication details (e.g., authors, year, title, source, abstract, aims);
- Context descriptions (e.g., subjects, technologies, industry, settings);
- Findings (e.g., results, behaviors, actions, phenomena, events, quotes)"

Cruzes and Dybå (2011) suggest presenting publication details and context descriptions in a tabular format. Therefore, I performed the data extraction of findings and qualitative coding of the selected primary studies. Finally, the results were elicited by aggregating and analyzing the coding of the papers. During data extraction, the process was discussed with

research colleagues to decide what to extract with relevance to impacts on teams in large-scale ASD work settings.

5.3.1 Paper inclusion criteria

I defined three criteria to guide the inclusion decisions: 1) large-scale Agile software development, 2) empirical studies on experiences or effects on teams, and 3) papers must be peer-reviewed, i.e. either published in a journal or proceedings from a conference. For a study to be included, it needed to be relevant to all three criteria. For exclusion, several reasons to exclude studies emerged iteratively during the abstract screening process. The reasons for exclusion could be grouped into four areas: 1) no descriptions of impacts on teams, 2) not in accordance with the definition of a large-scale setting, 3) theoretical contribution without empirical data (including literature reviews), and 4) studies not within Agile software development.

I discussed with colleagues whether to delimit the study by choosing between distributed or co-located large-scale Agile organizations because of some studies, e.g., Šmite et al. (2010), specifically investigate globally distributed software development. I chose to include both types of localization for two reasons: 1) Some studies do not clearly state whether the software development organization is distributed or whether the organization is merely located in several places, and 2) with many teams cooperating, it is not easy to define what "distributed" really means. For example, is it a distributed environment if half of the teams are located in another office in the same city? In another building on the same premises? On another floor in the same building? Due to these difficulties in identifying whether it is a distributed setting or not, I chose to include all studies.

5.3.2 Identification of studies

Systematic literature reviews in the field of Agile software development conducted by Silva et al. (2017) and Dikert et al. (2016) showed that the highest amount of search results came from the Scopus[1], IEEExplore[2] and ACM Digital Library[3] databases. I chose to search in these databases and also added Business Source Premier[4] based on recommendations from a library information specialist. I designed a search string based on

[1] https://www.scopus.com/
[2] https://ieeexplore.ieee.org/search/advanced/command
[3] https://dl.acm.org/
[4] http://web.b.ebscohost.com/

Population, Setting, and Outcome PSO (also known as a modified PICO) as recommended by Boland, Cherry, and Dickson (2017) for search strategies in systematic reviews. Unfortunately, finding suitable keywords for Outcome, such as ”impacts”, ”experiences” or ”effects” resulted in very few search results and since there is still a small number of studies on large-scale ASD, only search terms for Population and Settings were used as listed in Table 4.

Table 4. Systematic search strategy

Search strategy: terms related to modified PICO and numbers of articles per database	
Population	“software development” OR “product development” OR “system* development” AND
Setting	“Agile” OR “Scrum” OR “eXtreme programming” OR “Scaled Agile Framework” OR “Large-Scale Scrum” AND “large-scale” OR “scaling”
Outcome	N/A
First search (the 28th of March 2018): No. of articles per database *Range: 1995 – 2018*	Scopus 359 articles, IEEExplore 150 articles, ACM Digital library 106 articles, and Business Source Premier 56 articles (170 duplicates)
Second search (the 15th of July 2020): No. of articles per database *Range: 2018-2020*	Scopus 137 articles, IEEExplore 51 articles, ACM Digital library 344 articles, and Business Source Premier 13 articles (63 duplicates)

The results from the search process are displayed as a PRISMA Flow Diagram (Moher et al., 2009) in Figure 5. The database keyword search matched 671 papers, of which 501 were unique papers after removing duplicates. The resulting studies based on the database search were refined in two steps. First by screening all abstracts and, second, by screening the remaining full-text papers.

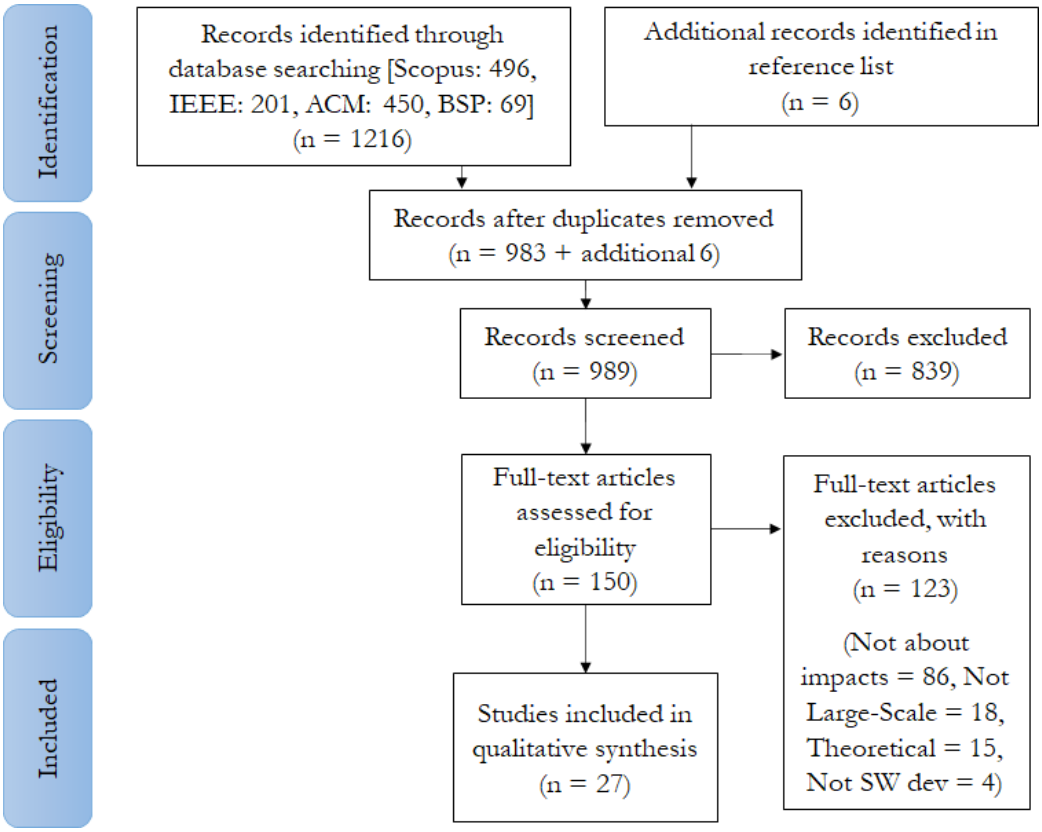

Figure 5. PRISMA Flow Diagram.

The manual screening of abstracts was executed by myself, but a small number of abstracts were independently screened by two other research colleagues. Different views of the relevancy of the abstracts were thereafter discussed, and decisions on inclusion resolved. This resulted in 150 articles left for full-text screening. Of the remaining 150 articles, I conducted full-text screening, and one included article was screened by both the two additional researchers and me. This article was discussed regarding its relevancy and quality to agree on how to evaluate the level of evidence and assessment rating. The screening was performed by evaluating the content of each paper against the three inclusion criteria. During the abstract screening process, papers, where one or two criteria could not be evaluated, were passed on to full-text screening. As a result of the full-text screening, only 27 papers were finally selected for inclusion, after meeting the three inclusion criteria (of which five of them where additional studies identified in the reference list).

5.3.3 Quality assessment

There are 11 questions for quality assessment in systematic literature reviews proposed by Dybå and Dingsøyr (2008). The questions are chosen from earlier versions of CASP Qualitative Research Checklist (Critical Appraisal Skills Programme, 2017), and are used to assign a value of 1 for "yes" and 0 for "no", to summarize an assessment score for each study. I adapted the sixth question, "Was there a control group with which to compare treatments?" into "Was there a control group with which to compare results?" since the original question was meant for health studies and does not fit well in this research area. Besides the 11 questions, I added two more questions from the CASP Checklist (2017) in two areas that were not part of the selection from Dybå and Dingsøyr (2008). The two questions, "Were recommendations made for policy or practice?" and "Have ethical issues been taken into consideration?" were therefore added. This amounted to 13 questions, and thus the maximum assessment score for a paper in this assessment is 13.

Besides the quality assessment, I used the GRADE system to decide the level (or strength) of evidence as recommended for SLRs in software engineering by Dybå and Dingsøyr (2008). Twenty-one papers had a clearly defined research method, but effects or impact on the teams were only a minor part of all the 27 selected papers. Based on this, I conclude that studies presenting impacts on teams in large-scale Agile organizations are scarce. I assumed that the results would be distorted heavily, and many valuable studies would be left out if a strict quality assessment was part of the inclusion criteria. As a result, I decided to include experience reports, regardless of the potential problems of author bias, thereby keeping all 27 papers.

5.3.4 Coding of primary studies

Cruzes and Dybå (2011) describe three coding strategies: deductive (a priori), inductive, or integrated approach. The authors further explain that "the integrated approach is the most relevant in systematic reviews as they tend to be done on the basis of the theoretical interests guiding the research questions of the review" (Cruzes & Dybå, 2011, p. 279). Therefore, I created a deductive start-up list of code types consisting of only two codes based on the intended areas to find, namely "benefits" and "drawbacks". Then, additional codes were developed inductively, along with the analysis, resulting in 69 more codes after the initial coding process. Four

articles were quantitative studies, but since the aim was not a quantitative meta-study, thematic coding was conducted on these articles as well by coding the presented findings. (In fact, these four studies did not measure similar areas, and a meta-analysis would not be possible to conduct.)

5.3.5 Synthesis of findings

Cruzes and Dybå (2011) recommend using thematic synthesis when conducting systematic literature reviews in the software engineering field. This follows four steps based on Braun and Clarke's (2006) method for thematic analysis:

1. Code data (interesting concepts, categories, findings, and results)
2. Translate codes into themes (themes, sub-themes and higher-order themes)
3. Create a model of higher-order themes (showing themes and relationships)
4. Assess the trustworthiness of the synthesis

Findings were synthesized by organizing codes into themes based on code labels. Each code was classified into a single theme. After assigning each code to a unique theme, the content of the themes was studied by reading through each quotation of each code included. Notes were taken on quotations presenting noteworthy observations, and the themes were refined based on these notes by changing definitions of some themes and re-classifying some codes until a final ordering was reached. The notes were also used to rename some of the themes based on reflections and discussions with colleagues about the content of these notes, quotes, and code names.

Evidence of benefits and drawbacks was mentioned a varying number of times in the articles, from only two up to fourteen. Braun and Clarke (2006) explain that frequency counts in thematic analysis is not mandatory but is sometimes used. However, merely the frequency of a mention of the impact is not sufficient as generalizable evidence for the actual importance of the impact. The result, a model of fifteen higher-order themes where five are benefits and ten are drawbacks, is presented in Table 2 in section 2.5.

5.4 Case study method

After finishing the literature review, which, as mentioned, provided a very small number of studies directly related to the core question of my study, I needed to decide on the proper methods to empirically investigate the

phenomenon of interest. The previous studies were not conclusive regarding the experienced impacts of implementing and tailoring inter-team coordination. Neither were they conclusive regarding details of tailoring.

If a researcher wants to gain an in-depth understanding of a real-life phenomenon, conducting case studies is the preferred research method (Merriam, 1988; Yin, 2017). Specifically, it is the preferred method when a researcher is seeking answers to *how*- and *why*-questions regarding a phenomenon. A case study approach was deemed appropriate in this study since the purpose of the research is to understand *how* inter-team coordination is performed. Another important purpose of the study is to understand *how* employees perceive the impacts of implementing inter-team coordination routines. The *why* is also important to understand: the reasoning behind why organizations tailor or do not tailor the studied coordination routines.

Case study research is also the most appropriate choice when a contemporary event over which the researcher has no control is studied (Yin, 2017). This method was therefore selected due to the difficulty of studying large-scale Agile development projects with other methods (such as conducting experiments), and due to its power to investigate new and interesting phenomena. Case studies are well suited for studying new and atypical phenomena. Such studies are sometimes called revelatory case studies: studies that reveal a phenomenon hitherto unexplored (Yin, 2017). Although there are previous studies on inter-team coordination (e.g., Dingsøyr et al. 2018; Nyrud & Stray, 2017), they are few, and more research on inter-team coordination is called for (Dingsøyr et al. 2018). Therefore, this is a case study of a revelatory nature.

The case study method does not require an existing theory or hypothesis to be applicable, and case studies can be theoretically generalized by using a replication logic. This means that large samples required for statistical generalization are not necessary (Yin, 2017).

In his sixth edition, Yin (2017) does not describe the process of doing case study research in five sequential steps like he did in earlier versions of his book. Instead, the author describes it as a linear but iterative process, as depicted in Figure 6.

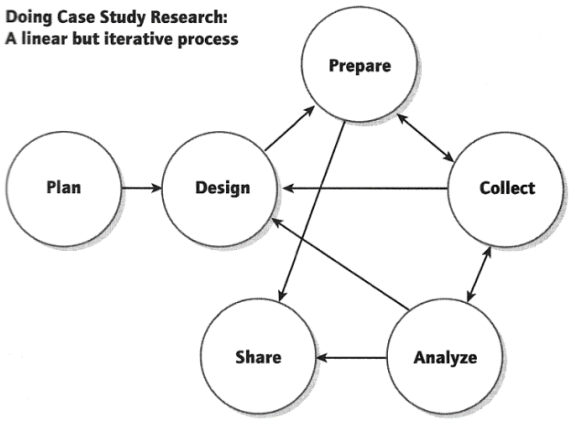

Figure 6. Case study research process (Source: Yin, 2017).

What Yin (2017) tries to explain in his new process view of case studies is that it is natural to iterate several of the steps along the way. This iterative way of working has been very much prevalent all through this study. However, as Figure 6 shows, the first two steps are to plan and design the study, which is where I started out for this study as well.

In order to plan and design this case study, I have studied and applied the suggested typology of the case study by Thomas and Myers (2015). The typology provides a systematic structure that facilitates how to clarify the purpose of the study, select the analytical approach to be pursued, and identify a proper process in conducting the study. The selected case study design for this thesis is presented in Figure 7.

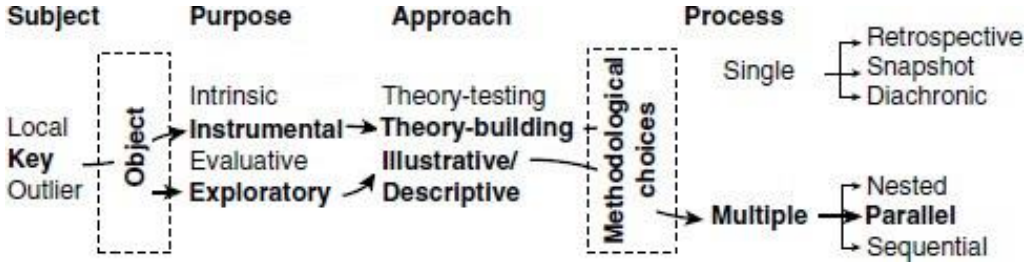

Figure 7. Case study design, according to Thomas and Myers (2015) typology.

Described according to the typology, the case study uses three *key* subjects, i.e., the three cases selected. The *object* of the study is to investigate the phenomenon of inter-team coordination routines in large-scale Agile software development projects. The key cases were selected with an *instrumental* purpose, i.e., cases chosen for a reason as opposed to an open-ended selection approach (Thomas & Myers, 2015). The purpose of the study, however, is of an *exploratory* nature. As such, a *theory-building* or theorizing approach was chosen. The research is also based on an *illustrative-*

descriptive approach since the study contains in-depth descriptions of the phenomenon.

A researcher can choose between a single or multiple-case study design. According to Yin (2017), the following five rationales are often a reason for choosing a single-case design: the case is critical, unique, representative, revelatory, or longitudinal. However, a multiple-case design is preferred over a single-case design because the conclusions may offer more possibilities to suggest generalizations. Actually, if a researcher can even do a two-case study, the chances of producing credible results will be better than using a single-case design (ibid.).

Because my aim is to understand the phenomenon of inter-team coordination from a holistic perspective, a multiple-case study design was the natural choice for my study. If I would focus on only one organization and study how they tailor coordination routines and the impacts of these choices, it might be difficult to transfer the results to other settings. Hence, I decided not to have a single case study design. The use of multiple cases allows for cross-case comparison and may give more robustness to the results of the study (ibid.). Multiple-case studies also make it easier to propose arguments (ibid.). However, by presenting multiple cases, there is a risk of only providing *thin* descriptions, rather than revealing deeper social dynamics that can be achieved by single case studies. I mitigate this risk by not choosing more than three cases and describe the studied phenomenon in much detail, providing thick descriptions.

As noted by Eisenhardt (1989), it is rarely possible to seek out ideal sites for case study research, and as such, there is a need to react to opportunities for intervention when and however they arise. I have been fortunate in finding suitable organizations at the very start of their upscaling journey, which was the cases I searched for. This will be further explained in the next section (5.5). Once again, to use the typology of Thomas and Myers (2015), a *multiple*-case study was chosen where cases were studied in *parallel* (different organizations during the same time period).

5.5 Case selection

Because I chose a multiple-case study design, the first question was how many cases I should study. I chose to study three sites for the following reason: Focusing on a few cases allows each one to be examined in greater detail. It is important to note that I do not systematically survey all pos-

sible organizations. Instead, I focus on a few cases with both similarities and differences in their plethora of products, services, and systems, sometimes referred to as a replication strategy (Yin, 2017). There is no reason to suspect that the sites I chose are atypical, but they are diverse. Nevertheless, this selection technique does suggest caution in generalizing my findings (ibid.).

I applied five criteria to my selection of cases:

Experienced in Agile ways of working. Many studies have been conducted focusing on the implementation of Agile ways of working in software development settings. Studies regarding how to implement, as well as the benefits and challenges of the use of Scrum and other frameworks, are also common. Since my interest is rather on challenges in scaling up ASD organizations that have already gained experience and are used to Agile ways of working, *experienced in Agile ways of working* was an important criterion.

Newly organized large-scale settings. Since the focus of my research is to study coordination between teams in a setting where many teams cooperate towards a common goal, the organizations needed to be large-scale. As previously discussed regarding how to define large-scale, I searched for settings containing 50 or more people or at least six teams cooperating towards a common goal. It was also important to find organizations at the very start of their implementations of new routines specifically considered for large-scale purposes to be able to collect data for a longitudinal study. That does not mean that the organizations needed to be newly formed, only that their implementation of inter-team coordination routines was recently started.

Variety of business logics. Varying the type of businesses in the organizations gives the cases a range of organizational structures, rules, and environmental conditions. This approach is what Yin (ibid.) refers to as choosing multiple cases that include contrasting conditions. To pursue this maximum variation approach, I studied organizations within technical product development and banking as well as a government agency.

Working according to the same framework. The ways organizations "are Agile" can differ in many ways, even if they work according to the values and principles of the Agile Manifesto (2001). A conclusion that can be drawn from Gustavsson (2017) is that roles and their responsibilities for coordination can differ vastly in different Agile settings. In order to have some coherence in what Agile means, it was important to find organizations with the same basic terminology and roles in order to make mean-

ingful comparisons on different ways to conduct the same described routines. Therefore, one criterion was to find organizations that implemented the same type of large-scale ASD framework.

Interested organizations. Obviously, an important pragmatic requirement was to find organizations that were interested in research and allowed me to observe their work. This criterion may introduce a selection bias since organizations interested in the study might also be more active and self-aware adopters of new management techniques. Companies that feel their processes work badly might be unwilling to be studied. Still, it would be difficult to pursue the intended research if the organizations were not willing to allow on-site visits, recording of interviews, or access to documentation.

The three cases Auto, Bank, and Gov, were chosen based on a maximum variation approach, selecting organizations that showed a variety in size, business area, and corporate culture. The cases are further described in chapter 7, 8, and 9, but here is a short introduction to the three organizations.

Auto (pseudonym) is a department responsible for product development of a significant part of a motor vehicle, both the software and the hardware. In 2013, the company started to implement Agile ways of working for software as well as hardware. The department was first organized into twenty cross-functional teams, and in 2016 they made their first attempts to implement inter-team coordination routines, mainly inspired by the SAFe framework. Team sizes spanned from two to thirteen team members. The twenty teams were organized into three release trains, and the first time I visited the company, they were conducting their third ten-week PI planning event.

Bank (pseudonym) is a department in one of the major business banks in Sweden. It consisted of six to eight teams that had worked together for less than a year and decided to implement inter-team coordination routines one month prior to my first visit. Team sizes spanned from five to eleven team members. The department introduced Agile ways of working in dedicated self-organizing teams some time during 2014. The first time I visited Bank, I met the Agile coach in charge of implementing SAFe when he was preparing for their first PI planning event.

Gov (pseudonym) was a pilot project at a large Swedish government agency that started with the aim of finding best practices for inter-team coordination routines to be used within the whole agency. Gov consisted of six to nine teams. Team sizes spanned from four to twelve team mem-

bers. A one-year review of the pilot project was intended to be able to provide guidelines for new rollouts of future large-scale Agile projects in the organization based on experiences of this pilot project. I had my first meeting with the management team during the preparation of the pilot project before their first joint planning workshop.

5.6 Data collection

Case studies can be performed in numerous ways, and for this study, I needed appropriate data collection methods to understand the practices and routines performed in the organizations. To prepare for data collection, I have followed the prescribed case study methodology suggested by Yin (2017) by developing a case study protocol (see Appendix C).

Pentland and Feldman (2005) describe that the performance of organizational routines is best studied by on-site observations too see what is done. I therefore visited the three organizations 17 times (379 hours), observing, making ethnographic field notes, and taking pictures with a digital camera. Figure 8 presents a timeline of on-site visits to the three organizations.

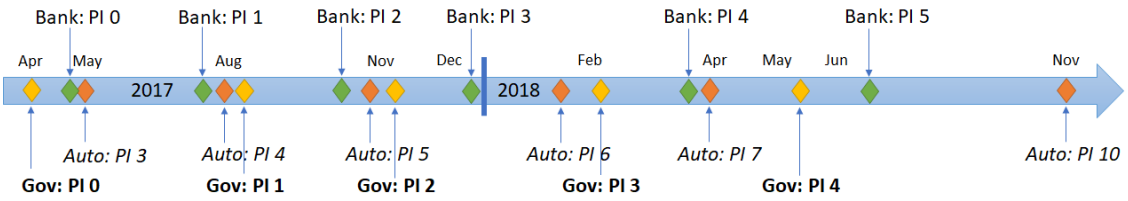

Figure 8. On-site observations at the case organizations.

I also participated in several online meetings (via Skype) and conducted 28 formal interviews. To understand *why* routines are conducted in a certain manner, the researcher needs to interview participants to hear the logic behind their reasoning (Pentland & Feldman, 2005). The roles of the interviewed respondents are presented in each chapter of the individual case studies, and the interview guide is presented in Appendix A.

According to Walsham (2006), interviews are the most important source of information but should ideally be supplemented, or triangulated, with other data, for example, data collected from participant observation and surveys. To get an understanding of the different aspects of how inter-team coordination was performed, I attended several PI planning events and Scrum of Scrums as well as other meetings to observe the routines and behaviors during typical workdays. Out of the four levels of par-

ticipation, according to ethnographical classifications by Gold (1958), what I did could be called a combination of *observer-as-participant* and *participant-as-observer*. On most occasions, I was typically a participant-as-observer being open and expressing why I was there and what my interest in the research was. In some situations, especially in large events, I was typically an observer-as-participant, since many attendees sometimes did not know what my role was or even that I did not belong to the organization. One could also say that some of the ethnographic data being collected was to some extent based on shadowing (Czarniawska-Joerges, 2007), since my main contact person at every case organization was one of the people in the Release Train Engineer role. To get access to Scrum of Scrums, to be allowed to attend PI planning events, I *had* to follow my contact person, who could allow me access to observe these routines.

Data collected through observations consisted of photos and field notes. I took more than 400 photos, but only a few of them were used for analysis, where I coded some of them. The main function of taking photos was to improve my field notes. An example: instead of writing down bullet points on a presentation slide used by a manager to present goals for a PI, I took a photo of the situation (with the presentation slide in the background). In my field notes, I wrote: "14.02. What? Other goals for implementing SAFe than presented in the last PI". Afterwards, by looking at the photo taken at 14.02, I could both see what the presented goals were and the role of the person presenting the goals. The case study protocol (see Appendix C) guided me in writing field notes and also provided advice on how to write ethnographical field notes (Emerson, Fretz, & Shaw, 2011).

Emerson et al. (2011) emphasize the importance of taking notes of the initial impression, which may include things available to the senses, such as the physical setting, equipment, number of people in the setting, sounds, and comments between people. This preserves the initial and often insightful impressions which might get lost, since observers "tend to lose sensitivity for unique qualities of a setting as they become commonplace" (Emerson et al., 2011, p. 24). The case study protocol helped me to follow another advice, which was to focus on "what is significant or unexpected" (Emerson et al., 2011, p. 24) to be able to document key events. The authors also suggest that the observer first register the feeling, then the event which took place. In my example above, the first word ("What?") was my way of expressing my feeling of being surprised. A reason for registering this feeling is to, later on, use this note to ask other

people in the setting how they experienced this event (Emerson et al., 2011). One way of understanding a phenomenon and what is important in it is to take notes on "what do they talk and gossip about?" (Emerson et al., 2011, p. 24). Therefore, much of my field notes contained sentences between attendees at the performed routines. I always tried to write down these dialog sentences consistent with the format of "when, where, and according to whom" (Emerson et al., 2011, p. 27). My field notes, therefore, contained a mix of initial impressions from events and dialog sentences.

I also used interviews to gather qualitative data and chose the *general interview guide* approach (Patton, 2002), also known as the semi-structured interview. I chose this form to be able to maintain adaptability to the roles and individual experiences of the interviewees while making sure that the relevant topics were explored. As March and Simon (1958) point out, "most programs are stored in the minds of the employees who carry them out, or in the minds of their superiors, subordinates or associates. For many purposes, the simplest and most accurate way to discover what a person does is to ask him" (p. 142). The first draft of the interview guide was based on Heikkilä (2015) who performed case studies on release planning in large-scale ASD. From this version, I updated the interview guide constantly based on observations and insights from the previous interviews. This also meant that I realized the need to adjust some words based on language use. For example, I described earlier why I use the term *routines* for inter-team coordination, but events such as PI planning and Scrum of Scrums were regarded as *processes* by the interviewees. Such adaptions were needed both for achieving a common understanding but also to save time by not having to explain why I used different terms than the ones used in the organization. Some interview questions were also based on the role of the interviewee. All interviews were voice-recorded, and each interview lasted between forty-five minutes and two hours and allowed for understanding the individual's views and pre-understanding.

Reading archival records, mainly memoranda from meetings, also helped to capture different elements of inter-team coordination routines. Observations and interviews in the various cases are presented in Table 5, and the roles of people interviewed are presented in each case chapter. Each on-site visit lasted for two to five working days. The longest visits where at Auto, since they were divided into three Agile Release Trains, and conducted PI planning one at a time for four or five days in a row. Data was collected from the starting point of implementations in the or-

ganizations (from January to April 2017) until February 2019, a period of almost two years.

Table 5. Observations and interviews.

Case organization	**# of on-site visits**	**Hours of observation**	**# of interviews**	**Hours of interviews**
Auto	6	196	14	11h 48 min
Gov	5	113	6	6 h 12 min
Bank	6	70	8	7 h 32 min
Total:	*17*	*379*	*28*	*25 h 32 min*

After 379 hours of observations and 28 interviews, theoretical saturation was considered reached, since few new insights about the tailoring of coordination routines were found. At this point, only repetitive data was being collected, and it was possible to provide "some direction for operationalizing" (Bowen, 2008, p. 140).

To understand the aspects of every whole case and to compare differences between cases, I decided that I needed a way to capture a larger population within each case, not to be biased by a few interviews. As Yin (2017) explains, when findings and interpretations are based on multiple sources, the data will be less sensitive to differences from one or two "outlier interviews". Although observations were used to achieve a more detailed picture of what was going on, the lack of personal cloning abilities caused me to only attend at one place at a time. Especially, I wanted to understand if the perceived impacts expressed in interviews were representative of the employees in each case. To be able to capture many opinions, I conducted a questionnaire-based survey which I used in all three case organizations. The survey included two open-ended questions:

1) "What do you consider as the main benefit of working according to SAFe in your organization?"

2) "What do you consider as the main drawback of working according to SAFe in your organization?"

Since these were open questions, some respondents gave several answers to these questions, respectively. What I really was interested in was to understand how the implemented inter-team coordination routines affected the team members, but I found it hard to pinpoint a question about only inter-team coordination that could be well understood. Instead, I chose the above formulations of "working according to SAFe" since that would be easily recognizable as the difference compared to the previous way of working with coordination between teams. This led to

some answers not entirely useful for my study with opinions not related to inter-team coordination. These answers were excluded from the study.

Paper-based questionnaires were used at all the organizations. They were handed out and collected during one of the PI planning events that I visited. The survey was conducted in February 2018 for Auto in March 2018 for Gov and in April 2018 for Bank, which means that the organizations had worked with the implemented inter-team coordination routines for roughly one year. Besides the two open-ended questions, I used the survey to collect some quantitative data as well. When I handed out the surveys, I was allowed fifteen minutes to explain how to interpret the questions in the different sections of the questionnaire and to inform the presumptive respondents that this was voluntary and anonymous. Thereafter the questionnaire was handed to the respondents.

The questionnaire consisted of multiple sections: (1) background (e.g., role and Agile experience), (2) opinions on working according to some of the routines in the SAFe framework, (3) perceived differences in teamwork quality between the previous way of working and current way of working, and (4) perceived view on the current inter-team collaborations and teamwork quality. The two open-ended questions described above were part of Section 2. A mix of negatively and positively worded items was used to mitigate response set bias. The survey questionnaire is presented in Appendix B. The collected quantitative data is mainly used as a complementary data source in this thesis. Only simple statistics, such as mean, median, mode, and standard deviations, have been used to bring further understanding of the cases.

To conclude on data collection, I have used multiple sources of evidence, and I have collected all evidence in a case study database (consisting of material stored in file folders as well as NVivo), thereby following the principles of data collection according to Yin (2017). To reduce researcher bias, a data triangulation approach was chosen. Interviews were conducted, on-site observations were performed, and questionnaire data was collected. Triangulation may yield more reliable data than the use of only one data source because what people report is not always consistent with reality. Interviews yield subjective data, but the more interviewees stating the same opinion, the less likely it is that bias is associated with that opinion (Diefenbach, 2009).

5.7 Analysis and presentation strategies

Data was analyzed using the approach known as *thematic analysis*, following the method proposed by Braun and Clarke (2006), which consists of six phases:

1. Familiarizing with the data
2. Generating initial codes
3. Searching for themes
4. Reviewing themes
5. Defining and naming themes
6. Producing the report

The first phase, *Familiarizing with the data*, consisted of reducing the observations notes, transcriptions, and textual documents into a shorter summary. This was conducted in two ways. First, all field notes written during each visit were summarized into a short (3-7 pages) narrative describing events chronologically. This summary also contained important quotes from discussions and meetings. Second, during each interview, notes were taken which were later used to make short summaries after each interview. These summaries were a way of trying to capture the most central information from each respondent and to get a better overview.

Thereafter, the interview transcripts and observation summaries were imported into the analysis software NVivo 12 to be used for analysis. NVivo helps to keep data organized in a case study database (Yin, 2017) and makes the process of coding and analysis transparent. Thus, using NVivo improved the reliability of the qualitative analysis. During this stage, all answers to the open-ended questions were also imported from the conducted survey into NVivo, along with some documents (such as meeting agendas and meeting minutes).

In the second phase, *generating initial codes*, I assigned initial codes to the data adhering to informant terms, which is also called *in vivo coding* (Miles, Huberman, & Saldaña, 2014). This coding of the data was conducted in three ways. One set of codes emerged based on data related to *how* inter-team coordination routines were performed and tailored. The second set of codes was based on data related to *why* the routines were chosen and tailored. The third set of codes was based on data about perceived impacts. While coding the data, some of the photos taken during on-site observations were also imported. The photos were coded in the same way as texts, by marking an area in NVivo and assigning it a code.

To give an example of the coding strategy, I have chosen to present one of the themes identified as a perceived drawback in the Gov case. Data was collected from answers to open-ended questions of perceived impacts, interviews, and observations. The observations were conducted when Gov performed inter-team coordination routines, such as during PI planning, Scrum of Scrums, and other meetings. To give an example of the first step of coding, Table 6 presents data supporting code generation. At first, codes were inductively created from answers to the open-ended questions. The specific question for the answers in this example was: *What do you consider as the main drawback of working according to SAFe in your organization?*

Table 6. Coding example using answers from open-ended questions.

Code	**Answers from open-ended questions**
Too many meetings & overhead	"Lots of meetings"
	"Too much overhead"
	"Time, cost of meetings"
	"Administration"
	"Too cumbersome"
	"I have to spend huge amounts of time on meetings that do not concern me"
	"There is far too much time for meetings/planning/follow-up"
	"Much overhead compared to our previous processes"
Code	**Answers from open-ended questions**
Inefficient	"Inefficient"
	"Inefficient"
	"It's too slow, not efficient"

Then, I created codes based on all transcriptions from interviews, observations, and meetings I attended. Table 7 on the next page shows examples of data supporting codes based on perceived drawbacks.

Table 7. Coding example using transcriptions from interviews and observations.

Code	**Quotes from transcriptions**
Approval culture	Interview: "…how it works at Gov, you have to get approval from above. Then it is full speed. But without approval, nothing happens, then everything happens." *Alina Coach*
	Interview: "this with the culture of approval, if no one has approved that you can stop doing something, then you continue doing it". *Alina Coach*
	Interview: "…we very often have to get things approved from other departments before we can build things." *Helena RTE*
	Interview: "often [the problem] is getting approval from someone or something. Some manager who has to have his say before we can move on." *Pierre SM*
	Scrum of Scrums: "Employees are afraid to make mistakes. There is a fear that there will be consequences if a decision turns out to be wrong." *Meeting the 6th of December 2017*
	Scrum of Scrums: "the problem is that we thought that Anna, who was the manager [of another department], should be able to make the decision, but she has no mandate. Since we deviate from the Governance model, it is the Director-General who needs to make the decision. She had no problems with that, but the formal decision has not been made yet." *Meeting the 13th of September 2017*
	Meeting with Product Owners: "Just before we went to [PI planning], we produced two examples and submitted to the legal department, which we have got some ... I should not interpret it as an OK, so maybe we can't plan for it either." *Meeting the 14th of September 2017*
Code	**Quotes from transcriptions**
Too much time in meetings	Interview: "all these meetings then. Meetings, meetings, all the time, long meetings where you feel like you are just wasting time." *Carl Dev*
	Interview: "We have spent far too much time in meetings, I would say" *Cecilia OpDev*
	Scrum of Scrums: "We have so many meetings. [Several voices of agreement]. Yes, there are many long meetings, a lot of people attending. Where do you take the money for that, or the cost of time spent in them, I mean." *Meeting the 13th of September 2017*

As the two tables above show, I chose to identify and form two sets of codes inductively. One set of codes based on answers to open-ended questions and one set of codes based on interview transcriptions. The reason for this was to avoid bias from the first set of codes (open-ended question answers) when analyzing the second part of the data. Also, there is a debate regarding whether answers to open-ended questions really could be considered qualitative research (e.g., LaDonna, Taylor & Lingard, 2018). However, open-ended questions could "help researchers

begin to sketch content areas" (LaDonna et al., 2018, p. 351). That is why I chose to start by analyzing open-ended questions and create initial codes based on merely this data. Another benefit of the divided data structure is that it further increases transparency of the origin of the data when presenting themes, as can be seen below in Table 8.

The decision to create themes in an inductive manner based on two data sources took some time to come up with. One earlier attempt to analyze benefits and drawbacks can be seen in Gustavsson and Bergkvist (2019), where the answers were instead deductively organized into themes, or key areas, based on the agility evaluation grid (Laanti et al., 2011). Although the paper contributed with insights into differences between my studied cases and the case study at the company Nokia, it did not describe the studied benefits and drawbacks in detail. In this thesis, I decided to present thicker descriptions, and deeper insights, by creating themes inductively based on data from several data sources.

During the third phase, *searching for themes*, I started to look for similarities and differences among the many codes from both sets of codes. In this phase, the previous codes could also be combined, divided, or even withdrawn. By identifying patterns, the first themes and sub-themes started to emerge (Braun & Clarke, 2006). A theme should capture something important about the data in relation to the research question (ibid.). The idea behind the two coding steps is to identify patterns and find broader underlying themes that connect codes. Yin (2017) suggests identifying theoretical propositions as the basis for analyzing data. By adopting a pattern-matching logic, comparing empirically based patterns with the theoretical propositions, internal validity is reinforced (Yin, 2017).

To identify themes on *how* routines were performed, a deductive approach was applied. Concepts from coordination theory guided the analytical procedure, such as different types of dependencies, emerging versus planned coordination, and coordination mechanisms (Malone & Crowston, 1994). I published a first version of this analysis at a conference (Gustavsson, 2018). Based on the feedback and discussions at the conference, I analyzed the data further and revised some themes into the results presented in this thesis.

The same deductive approach was used for identifying themes on *why* routines were performed and tailored. Concepts from institutional logics were used, such as the different ideal institutional order types and categories identified, developed, and presented in chapter 6.

To identify and build themes on perceived impacts, an inductive approach was instead applied. First, I searched for evidence of perceived benefits and drawbacks, and then I used previous findings on impacts identified in the systematic literature review. All themes were examined to make sure they were distinct and separable, that they contained coherent data, and that they represented the entire data set, in accordance with Yin (2017).

Table 8 presents an example of an identified theme based on perceived drawbacks in the Gov case.

Table 8. Example of the data structure for a theme of perceived drawbacks.

Codes from open-ended questions	**Codes from observations and interviews**	**Theme**
Too many meetings & overhead [8]	Approval culture [7]	*Too much time spent in meetings*
Inefficient [3]	Too much time in meetings [3]	

As discussed above, keeping the two sets of codes separate based on the origin of the data makes it fully transparent when presenting the themes with underlying codes. The numbers within brackets in the table are used according to the same principle as in Frith and Gleeson (2004) to present the number of data units relating to each code. As previously mentioned, there is some criticism against using answers to open-ended questions as qualitative data (LaDonna et al., 2018). The claim is that answers to open-ended questions do not provide as much in-depth information as verbal interview answers. However, critics still mention the value of using them when "data were purposefully selected to answer the research question" (LaDonna et al., 2018, p. 353). In this case, the open-ended questions were purposefully selected to answer the research question on perceived impacts. Also, answers from open-ended questions were not the only source of data, which I argue mitigates the potential lack of depth in answers from open-ended questions.

To analyze how team autonomy was changed in the cases (to be able to answer RQ4: *How is team autonomy perceived to be changed after implementing inter-team coordination routines?*), I chose a different approach. The data used for analysis were interview answers to questions specifically regarding perceived changes to team autonomy. I used the Agile principles as stated in the Agile Manifesto (Beck et al., 2001) with an analytical purpose. I analyzed the perceived changes to team autonomy deductively based on whether they supported or conflicted with the Agile principles.

In the fourth phase, *reviewing themes*, it became evident that some candidate themes were not really themes. Some were too diverse, while others collapsed into each other (Braun & Clarke, 2006). Braun and Clarke (ibid.) describe this phase as action on two levels: first, to look at the coded data extracts for each theme and a second level to map all themes with the entire dataset. For me, this phase was very much an iterative process of jumping between the two levels. Once I thought I had recognized the proper data extracts for one theme, I understood from a holistic perspective that this theme was either too broad or too narrow. I realized that the problem with some of my broad themes was that they became too vague and bland. Some of my narrow themes were too detailed to understand why they were important. All through this phase, I used a semantic naming approach to the themes trying not to lose the voice of the informants. The semantic naming of codes and themes was very helpful during the refining of themes to better understand and remember which codes that belonged to each theme. This work was carried out for several iterations, and, as Braun and Clarke (2006) warn about, it is hard to know when to stop.

For the fifth phase, *defining and naming themes*, themes were named and rewritten by following Alvesson's (2011) notion of 'reflexive pragmatism'. Reflexivity means "working with multiple interpretations in order to steer away from traps and to produce rich and varied results" (Alvesson, 2011, p. 106). I tried to embrace reflexivity by challenging and questioning my interpretations of observations and quotes to be able to define each theme and give them distinctive and representative names. As Braun and Clark (2006) explain, themes need to be concise, punchy and give a direct sense of what the theme is all about. I also used the test encouraged by the authors to see whether I could describe the scope and content of each theme in a couple of sentences.

5.7.1 Presenting findings from each case

For the sixth phase, *producing the report*, I looked at examples of case study papers from both IS research as well as Software Engineering, where many papers on ASD can be found. As van der Blonk (2003) points out, it is surprising that the ways or the forms in which case study research can be written and reported are rather neglected in much case study methodology literature. In his paper, van der Blonk (2003) suggests different forms of writing down case studies for IS research based on decisions along two dimensions. The first dimension to decide on is reduction ver-

sus complexity. It refers to the level of complexity that the researcher wants to portray in the case study and is a split between attempting to understand the case through reduction versus attempting to understand the complexity of the case. One end of the spectrum means aiming to reduce the richness and variety of the case to a predefined format (such as chronology or a storyline) so that a certain pattern or insight is generated. At the other end of the spectrum, the researcher explains the complexity and variety of the case study explicitly by including contradictory material, chaotic situations, or emotions.

In writing about my case studies, I first chose the reduction approach to depict the longitudinal observations chronologically in a classical three-act-format (ibid.). This was to give a descriptive and illustrative view of *how* routines are performed and tailored. The previously identified themes were used to guide the process of writing down the chronological parts of the case descriptions. In the later part of the case presentations, I chose the complexity approach to present different opinions, views, and perceived impacts. The idea behind this approach was to offer a deeper understanding of *why* routines were chosen and tailored. Coordination theory and new institutional theory were used to guide the process of writing down these later parts of the case descriptions.

The second dimension to decide on was whether to describe the case from a monovocality or multivocality perspective (ibid.). The monologue end of the spectrum means that a case is presented as a uniform story that is told in the author's way. The other end of the spectrum means that the case is presented as an interaction of multiple voices, possibly including the author's voice as one of them. All through the case presentations, I use a multivocality perspective where the researcher's voice is only one of the voices present in the case descriptions. The reason for this choice was to be more transparent. Instead of only presenting from my perspective, using actual photos and quotes from different actors in the organizations increase transparency and thus construct validity (Yin, 2017). Also, the quoted phrases and sentences help to present the participants' perspectives and thinking (ibid.).

5.7.2 Cross-case synthesis

Cross-case synthesis aims at bringing together the findings from separate case studies and is the most critical part of a multiple-case study (ibid.). There are several common problems in multiple-case studies. One such problem is that multiple-case studies have difficulties in comparing cases

from diverse contexts. Studies sometimes focus too much on the special characteristics of each case and try to answer the research questions through single studies (ibid.).

The cross-case synthesis in this thesis aims towards a knowledge support goal and not a decision support goal (Pope, Mays, & Popay, 2007). A synthesis directed at knowledge support will typically bring together and synthesize research evidence on a particular topic. The goal is to create new knowledge on the topic. A decision support goal, on the other hand, must also include some or all analytical tasks required to reach a decision in a particular context (Pope et al., 2007).

I use the method called *cross-case analysis* with a *case-oriented approach* (Miles et al., 2014). The case-oriented approach is preferable when investigating a few cases in depth, while the variable-oriented approach is more appropriate when broad patterns are searched for in a large number of cases. Since I study only three cases, I decided to use the case-oriented approach instead of the variable-oriented.

There are different possible analytical strategies to choose from when a case-oriented approach has been chosen. Yin (2017) proposes the use of a replication strategy where one case is studied in-depth, and successive cases are examined to see whether a pattern found matches that of the previous cases. According to Miles et al. (2014), many researchers choose a comparison strategy by forming types. These types are arranged based on identifying whether cases fall into clusters that share certain patterns or configurations.

Sometimes, cases are more or less comparable and structured in a similar way. For these occasions, a meta-summary or meta-synthesis strategy can be used to systematically synthesize interpretations across several cases. My strategy is to conduct a meta-synthesis and, to some extent, a comparison strategy by clustering. As recommended by Cruzes et al. (2015), I adopt more than one method approach of synthesis for the sake of reliability of the results and conclusions.

When it comes to comparison strategies, I argue that the replication strategy is more useful when the cases studied only contain minor differences. It is then more fruitful to investigate additional cases one by one to analyze these small differences. Because I chose cases with a maximum variation approach (Yin 2017), I instead expected rather comprehensive differences between the cases. That is why I chose a comparison strategy of meta-synthesizing to display differences between all three cases.

The approach I have chosen is to partition and cluster data from the three cases to present several matrices so that contrasts between cases become clearer. This requires a further transformation of the data from the three cases into short quotes and summarizing phrases (Miles et al., 2014). These matrices help in visualizing the data of the individual cases but also present similarities and differences between them.

5.8 Validity and reliability

This section is divided into two parts. First, the presented research in this thesis is discussed based on four different aspects: construct validity, internal validity, external validity, and reliability. In the second part, I present my own background and highlight risks of possible personal biases.

5.8.1 Four aspects of validity and reliability

Construct validity is concerned with how well the constructs (concepts, themes, categories, and hypotheses) created during the analysis process reflect reality (Yin, 2017). The main way to improve the construct validity of a case study is to triangulate data sources, investigators, theories, and methods (Patton, 2002; Yin, 2017). In this thesis, data sources, investigators, and methods (or frameworks) were triangulated in all three cases. Triangulation of theories was not used, because the research was exploratory and descriptive, not theory-based and deductive. One threat to construct validity is that subjects under study may have reacted to the presence of the researcher and provided data that gave a biased image of the case organization or themselves. Thus, the results may not fairly represent their normal conduct and the reality in the case organization. The risk of this was mitigated by the longitudinal nature of the case studies. Over time, the employees at the case organizations got used to the presence of me. Also, triangulation was conducted by collecting data from many different sources and by using multiple data collection methods such as observations and interviews with people with different roles and from different parts of the organization. Another threat to construct validity is the inadequate identification of constructs. The constructs identified in this research may not fairly represent reality due to errors in the coding process or because of researchers' bias in the analysis. This threat was mitigated by investigator triangulation, since colleagues at Karlstad University and experienced members of the case organizations reviewed the results before publishing at conferences, where other experienced researchers reviewed the results as well. Also, when visiting the different case organi-

zations, I was often asked about what observations I had made. On several occasions, I presented parts of what I had observed. I knew that by presenting findings, there was a risk that I as a researcher would interfere in daily decisions. Therefore, I tried to present observations without expressing an opinion on whether what I had observed was good or bad. This is not only a possible weakness but also a strength. As Yin (2017) describes, a good strategy for improving construct validity is to let key informants review drafts of the case study report. To apply this strategy, I did not only present observations but also collected feedback on what I had observed.

The internal validity of a descriptive case study is concerned with the validity of the causal inferences identified in the analysis (ibid..). The identified causal relation may be temporally ambiguous; that is, it may be uncertain what is the cause and what is the effect. Events that happen concurrently or changes occurring naturally over time may create an illusion of a cause-effect relationship. In the cases, the analysis of cause-effect relations was limited to the cause-effect relations identified by the respondents. To understand the actual cause-effect relations, further studies will be needed, and such evidence will not be touched in this thesis. As Yin (ibid.) explains, internal validity is mainly a concern for explanatory case studies, not exploratory studies such as this one.

The external validity of a case study is concerned with the domain the results can be generalized to (ibid.). Threats to external validity arise from aspects of the cases that are not identified or disseminated, but that affect the results. It is often difficult to tell how much can be generalized from any particular case study. In single-case-study research, according to Yin (ibid.), the generalizability of the results to other domains can be considered only on a theoretical level. In my study, I address this concern by studying several case sites and comparing my findings across them, thereby increasing external validity.

Reliability of research is concerned with the repeatability of research (ibid.), i.e., if other researchers had conducted the same study, would they have reached the same results? Case study research is very much based on the cognitive skills of the researcher. The main threat to the reliability of the studied cases is variability in data collection. To mitigate this threat, many different data sources, a wide variety of interview questions, and several interviews were used, making the results more reliable (ibid.). The study was conducted following a case study protocol (see Appendix C). Following a protocol is "a major way of increasing the reliability" (ibid., p.

96) of a case study, especially for multiple-case studies. Also, in presenting the cases, a multivocality perspective using quotes and photos was a strategy chosen to mitigate problems with reliability.

5.8.2 Researcher bias

I have been working as a practitioner within software development in different roles, often as a project manager, since 1996. I have been teaching at Karlstad University on a part-time basis since 2003 while working as a consultant and author of textbooks on Agile project management.

Hence, researcher bias might be a risk due to my background. As a consultant, I have been an advocate for Agile ways of working and have helped customers to introduce new working methods. This might have shifted my perspective towards an overconfidence in how Agile methods can improve work results in businesses. There is a risk that my practical experience excludes explanatory models other than those that agree with my own view of the world. To mitigate the risk or researcher bias, one strategy was to triangulate data collection in three different steps. First, I focused on observations. From the first number of observations, I revised the interview protocol based on my observations and continued to update the protocol based on each performed interview. The third step was to conduct surveys with each case organization where I could collect the perceived experiences from a very large part of the studied participants.

At the same time, my background gives me advantages in studying a phenomenon where I have much experience. One advantage is related to the question of whether other researchers would have reached the same results based on the same data (Yin, 2017). I believe that my background may have had a positive effect on the data I was able to collect. Since I have first hand experience from the daily work in an environment close to that of the respondents, I could ask more detailed follow-up questions. Showing a detailed understanding of the respondents' work situation, legitimized me as a listener. This legitimization of me as an understanding listener, could be a reason why respondents were so willing to share their personal experiences with me.

5.9 Research ethics

The Swedish Research Council (2017) published a report on good practice for research ethics as a guide for researchers, especially for those at the beginning of their careers. The report stated that the protection of individual privacy is fundamental to scientific research in the humanities field

and described several guiding ethical principles. A few general rules for good research ethics are described in the initial summary of the paper, and I will use three of them, the ones most applicable for me, to discuss ethics relating to my research.

"1) You shall tell the truth about your research" (Swedish Research Council, 2017, p. 12)

The researcher should inform the persons concerned about the purpose of the research (ibid.). I was allowed a couple of minutes for a brief introduction of who I am and my research interest during the first visit to each of the case organizations during the PI planning events. I did not go into further detail after stating my interest in the scaling-up process of Agile work methods in general and inter-team coordination in particular. At the time of handing out my questionnaire during a PI planning event, roughly one year after the first visits, I was once again allowed a few minutes for a short presentation. The reason for the second presentation was above all that new people had started and did not know who I was. It was also an opportunity to present some information about the survey questionnaire and the intended research.

Besides this information for everyone, I had a designated contact person at each of the case organizations, which I had long discussions with regarding my research interest. This was also the case when conducting the interviews; the first minutes were spent giving more details of my research.

"3) You shall openly account for your methods and results" (Swedish Research Council, 2017, p. 12)

With both my contact persons as well as interviewees, I presented both the purpose of the research and how the publication of data and results were made. Consent means that participants in a study have the right to decide on their participation (ibid.). I stated that information from the interviews could be used in this thesis and in other research publications and I then asked each respondent whether she or he had any objection to attending. I received written consent from all respondents and gave them my contact details with an invitation to contact me regarding questions or to supplement or change any information in the interview.

During the hand-out of the questionnaires, I pointed out that participation was voluntary and presented contact information both on a presentation slide as well as on the paper questionnaires themselves.

"6) You shall keep your research organized, for example through documentation and filing" (Swedish Research Council, 2017, p. 12)

Confidentiality in research ethics means that participants are given the highest possible confidentiality and that personal data are stored so that unauthorized people are unable to access them (ibid.). Whenever I discussed or shared data with colleagues, all data was anonymized. Personal data about the interviewees was stored in a separate document that only I had access to. In this document, I also kept the key to the pseudonyms used in the other research material and in the thesis. The corresponding anonymization has been made regarding participants' organizations as well. I was inspired by the approach used by Jansson (2015) in his doctoral thesis, where respondents were anonymized using pseudonyms to give anonymity, but withholding names that give information about the roles of the respondents.

At Karlstad University, all research projects must be reviewed by an ethical advisor to check whether there are ethical implications in the conducted research. The research design for this thesis study has been reviewed and approved (registration number HS 2019/507).

6 Institutional logics in large-scale ASD

New institutional theory with ideal types and logics is described in chapter 4. In this chapter, I will use new institutional theory to conceptualize institutions and institutional logics in large-scale ASD. I suggest, in the same manner as Ferrer Conill (2018), that adopting an institutional logics perspective allows for an explanation of individual and organizational agency. In the following sections, I will propose two institutions, each with two opposing dominant institutional logics to be used for analysis in large-scale ASD.

The concept of software development as an institution is not new. Avgerou (2000) argues that the prevalence and use of IT in organizations has made it an institution in its own right. Rowlands (2008) further explains that the structures embedded within the methods for systems development, such as authority, norms, symbolic values, and routine ways of doing things, constitutes software development as an institution.

The concept of software development as an institution is not much developed and, as presented by Rowlands (2008) explained as only one single institution. Doležel (2018) developed the concept by dividing software development into two disparate institutional logics: Traditional Software Engineering logic and ASD logic.

One could suggest large-scale ASD as a third institutional logic, but I believe there are more nuances to large-scale ASD and that describing it as a single logic does not help us in unlocking the detailed insights of large-scale ASD. Therefore, my theoretical proposition is that there are two dominant institutions in large-scale ASD (*Agile ways of working* and *product development*) each with two disparate logics (*Agile rulebook logic*, *Agile toolbox logic*, *resource efficiency logic*, and *flow efficiency logic*), which are of importance for analytic purposes. It is not improbable that there are more than these two, but for the level of analysis I have chosen, I suggest the two institutions presented in the following sections to be most prominent.

6.1 Agile ways of working as an institution

The Agile ways of working, no matter which methods or frameworks adopted, are comprised of several defined roles, routines, and actions. The Agile Manifesto (Beck et al., 2001) was the starting point in the IT industry, but the Agile ways of working are being adopted in other industries as well (Gustavsson & Rönnlund, 2010; Gustavsson & Rönnlund,

2013). This means that Agile ways of working is more than Agile software development. I argue that it is an institution since it contains symbols, roles, routines, and actions taken for granted, which fits the definition of an institution (Ocasio et al., 2017). It can be considered an *informal institution* (North, 1990) which is defined as an institution loosely formed by interested actors and is more or less explicitly stated by the concerned agents. Although the Agile ways of working as an institution is an unexplored concept, scholars such as Doležel (2018) and Berente, Hansen, and Rosenkrantz (2015) investigate the logics of ASD, but without defining it as an institution as such.

Doležel (2018) argues that the advent of the Agile Manifesto (Beck et al., 2001) can be seen as the emergence of an identity movement where activists create symbols and terminology for a new institution. The Agile framework Scrum, Doležel (2018) explains, clearly distances itself from the current logic of traditional software engineering by expressing boundary markers, thereby suggesting an institution on its own. Another example is Berente et al. (2015), who investigate three ASD projects in different contexts. The researchers identify three different institutional logics, based on Agile ways of working as an institution separate from traditional software development. Berente et al. (ibid.) used differences in the corporate context to identify these three competing logics within Agile ways of working as an institution. To use the classification of institutional order ideal types (Thornton et al., 2012), this is an example of differences in *corporate logics*.

Berente and Yoo (2012) defined a number of institutional logics at play at NASA within software development such as *The Logic of Project Management Professionalism*, and *The Logic of Scientific Professionalism*. These logics are based on employee roles in the organization. In other words, they are examples of differing *professional logics* (Thornton et al., 2012).

Thornton et al. (ibid.) explain the importance of communities in organizational research and put forth *communities* as the seventh institutional order ideal type. This ideal type explains the strength of such social movements in which organizations sometimes reside in the shadow of communities as opposed to vice versa (O'Mahony & Lakhani, 2011). The community institutional order competes with or compliments other orders of the interinstitutional system, such as corporate and professional institutional order (Thornton et al., 2012). The authors argue that scholars historically have underestimated the importance of community logics

since this is essential to the evolution of organizations as mediators of performance and growth.

One such movement is the open-source community within software development, presented in Thornton et al. (ibid.). The open-source community is a social movement with a set of values, routines, symbols, and actions taken for granted, in accordance with the definition of an institution. The open source community is strong, both within companies such as IBM (O'Mahony & Lakhani, 2011), but also outside firms such as the community of software developers improving the free operating system Linux in their spare time (Thornton et al., 2012). There are also examples of developers who contribute to the free-of-charge open source community at work. For example, some employees at Google and HP contribute software code to community projects while simultaneously working towards firm goals (O'Mahony & Lakhani, 2011). These are examples of dominating community logics loosely coupled with competing logics such as corporate or professional logics at play. I argue that, just as open source community is an institution, the Agile ways of working is also a community with its own set of values, routines, symbols, and actions taken for granted.

6.1.1 Agile rulebook logic

My theoretical proposition is that there are two diverse logics, based on the community institutional order, in the Agile ways of working institution that is important to define. I choose to call the first logic *Agile rulebook logic*. This logic means that a proper Agile method is chosen and implemented in full as prescribed. Šmite et al. (2010) calls this the *all-or-nothing* attitude. Some advocates of this approach are Schwaber and Beedle (2002) and Jeffries, Andrewon and Hendrickson (2000). The method description becomes a rulebook, guiding the organization in how to implement the method. The commitment to the method descriptions becomes the *source of authority* (see Table 9).

The rationale behind many decisions is often based on the method description in this logic, viewing the prescribed descriptions as rules (Cao & Ramesh, 2007). A *root metaphor* for this logic would be: Agile routines as rules (see Table 9).

Method implementation and tailoring have been much studied in IS research (Fitzgerald, Russo, & O'Kane, 2003). Two various approaches to tailoring are often discussed. One is the *contingency* approach, which means that an organization maps a number of contingency factors with possible

software development methods and selects the appropriate method, or methods, to implement (Salameh & Bass, 2020). The contingency factors approach assumes that there is no sufficient method for all contexts. Hence, the right method should be selected based on the targeted development context (Fitzgerald et al., 2003). The tailoring is therefore based on which method to choose, rather than to tailor the method itself. This is the *basis of strategy* for the *Agile rulebook logic* (see Table 9).

Early advocates of Scrum (Schwaber & Beedle, 2002) argued that Agile methods cannot be applied *à la carte* but must be applied in their entirety to achieve the desired effect. One might consider this argument contradictory to the spirit of flexibility in Agile methods (Karlsson & Ågerfalk, 2009). Also, studies show (e.g. Fitzgerald, Hartnett, & Conboy, 2006) that the desired effect is possible to achieve even when only some routines are implemented.

However, Schwaber (2006, p. 1) also argues for another reason by stating that if "an enterprise modifies or only partially implements Scrum, it is hiding or obscuring one or more dysfunctionalities that restrict its competence in product development and management". In other words, rather than risk not achieving the desired effects, a problem of excluding routines is the possible loss of transparency, and the organization may not discover important problems. When asked by media about what an organization should be allowed to change in Scrum, Schwaber answered: "the length of the sprints, but nothing else" (Larsson, 2008). This answer, that Scrum must be applied in its entirety, is somewhat at odds with the intended flexibility expressed in Agile values and principles (Ågerfalk & Fitzgerald, 2006).

What Schwaber (2006) means is not that Scrum cannot be tailored, but rather that exclusion of parts of the method is not the proper way to do it. The originators of several Agile methods emphasize on describing *just enough method* to be able to be implement it in its entirety (Karlsson & Ågerfalk, 2009). Therefore, the *Agile rulebook logic* does not necessarily mean that tailoring is something bad, but rather that extensions could be added instead of parts of the chosen method being excluded. After all, theory on method tailoring suggests that no matter how well designed, there is no method that provides an exact fit for the needs of every project (Iivari, 1989).

6.1.2 Agile toolbox logic

The next logic, which I will define, I chose to call the *Agile toolbox logic*. This is also based on the community institutional order. This logic means that Agile routines are selected and tailored based on challenges in the ASD organization. Šmite et al. (2010) calls this the *à la carte* approach.

In early IS research, this approach to tailoring is called *method engineering*. The origins of the concept of method engineering can be traced to mechanical engineering in the 1930s (Fitzgerald et al., 2003). The method engineering approach implies creating a new software development method based on method fragments from defined methods and frameworks (Salameh & Bass, 2020). The strategy is to construct a situationally appropriate Agile software development method out of existing, proven method fragments (Conboy & Fitzgerald, 2010). This is the *basis of strategy* for *Agile toolbox logic* (see Table 9). The in-house creation of the method makes the team and the organization the owners of the method. The commitment to team decisions on tailoring becomes the *source of authority*, rather than method descriptions (see Table 9). According to a systematic literature review on reported Agile method tailoring, the *method engineering* approach is the most common one (Campanelli & Parreiras, 2015).

In early studies of the Agile methods XP and Scrum, McBreen (2003) and Fitzgerald et al. (2006) argue that parts of the Agile methods can be cherry-picked, ignored or replaced. These conclusions are presented as counter-arguments to Schwaber and Beetle (2002), who claim that the synergy of the entire Scrum method is greater than the sum of its parts and that most benefits are achieved through the combination of all roles and routines. Table 9 describes the main dimensions of the two ideal types of institutional logics based on dimensions presented in Thornton et al. (2012) and Thornton and Ocasio (2008).

Table 9. Ideal types of Institutional Logics in Agile ways of working.

Dimension	**Agile toolbox logic**	**Agile rulebook logic**
Root Metaphor	Agile routines as tools	Agile routines as rules
Sources of Legitimacy	Unity of improvement needs	Unity in interpretations, standardization
Sources of Authority	Commitment to team decisions	Commitment to method descriptions
Sources of Identity	Attentiveness and responsiveness	Method knowledge
Basis of Attention	Routine inefficiencies	Level of method adoption
Basis of Strategy	Implement, tailor and/or invent routines	Implement the method of choice

Although these are two competing logics with diverse approaches, they might also coexist in one organization. One example from early research on Agile method tailoring at Motorola is presented in Fitzgerald et al. (2003). An overarching method was first established within the company, but for each development project, micro-level tailoring of routines was conducted to suit the context.

Another example is highlighted in Salameh and Bass (2020), investigating the company Spotify which blends the two logics into what the authors call a *heterogenous tailoring approach*. The approach means that, first, each team is empowered to select and tailor its own method. Thereafter, each team is aligned to the overall product development goals through a number of added joint practices. This coupling of logics is not uncommon (Thornton et al., 2012).

6.2 Product development as an institution

Product development can be defined as a "network of interrelated activities that interact in order to transform a market opportunity into a product that meets the customers' needs and the strategic goals of a company" (León & Farris, 2011, p. 29). Product development has always been in need of increased effectiveness and efficiency to produce better and less costly products (ibid.). Also, product development projects often involve greater amounts of innovation, creativity, and iteration than many other types of projects (Browning & Ramasesh, 2007).

I propose that product development is an institution, since it contains symbols, roles, routines, and actions taken for granted, which fits the definition of an institution (Ocasio et al., 2017). Browning and Ramasesh (2007) describe how many scholars have defined how product development projects differ much from other types of projects and present in detail the specific processes and routines involved in product development. The researchers do not define product development as an institution as such but describe the specifics which span over more than only one corporate culture, and become a type of corporate logic.

6.2.1 Resource efficiency logic

I propose that there are two diverse logics, based on the corporate institutional order, in the product development institution that are important to define and use for analytical purposes. The first I choose to call the *resource efficiency logic*.

Resource allocation and utilization are crucial to product development projects (Joglekar & Ford, 2005). The resource allocation practice in many projects is based on an approach focusing on individual resources. Managers look at the current work to be performed and allocate resources to each activity in the same proportion that the activity's backlog contributes to the total amount of work waiting to be done (Joglekar & Ford, 2005). This traditional view on development work has focused on the flow of work that moves from one individual to another. *Resource efficiency* means that tasks are optimized for individuals (Tay, 2016). Resource efficiency refers to "the best use of resources in production" (Shone, 1981, as cited in Tay, 2016, p. 1159). According to this logic, efficiency is measured by the amount of output that can be gained from a certain level of input. The aim is to increase productivity through areas such as cost reduction, time management, and quality control. By increasing output and/or reducing the input, overall profitability is improved (Tay, 2016).

High resource utilization is the main goal in many organizations (Joglekar & Ford, 2005). Redundancy is often viewed as waste and, according to resource efficiency logic, a well-managed organization is, therefore, without excess capacity. This view of efficiency based on maximizing resource usage is embedded in many organizations (Tay, 2016). This sometimes encourages a silo view and a localized approach to process optimization at the departmental or sub-unit level (Coelli et al., 2005).

6.2.2 Flow efficiency logic

The second logic I will define, based on the corporate institutional order, I chose to call the *flow efficiency logic*. The logic of flow efficiency is to focus on value-adding activities with respect to throughput time. The question, at any given time, is whether the product being developed is receiving value or whether it is 'standing still' (Tay, 2016). To improve efficiency, the organization measures the time it takes a product to move through all the development processes.

A flow efficiency logic view suggests that a resource efficiency logic approach generates lots of waste and hidden inefficiency that results in a far lower overall efficiency than the company thinks (Reinertsten, 2009; Tay, 2016). The flow efficiency approach has also been called New Product Development (NPD), and highlights the value of a more structured way of reducing non-value-added activities but also the need for an emphasis on creativity to enhance value creation (Machado, 2013). Hence, according to flow efficiency logic, the organization needs redundant re-

sources, i.e., that all personnel is not fully allocated. Redundancy is important to allow time for creativity and innovations as well as 'extra resources' to help out if the flow is hindered (Machado, 2013; Tay, 2016).

The SAFe framework emphasizes flow efficiency as an important principle to get the most benefits out of SAFe (Scaled Agile, 2020). Organizations embracing the framework should be lean, according to SAFe, always striving to achieve a continuous flow of value.

Flow efficiency places responsibility for processes and tasks in the hands of teams rather than individuals. If one person is not present for any reason, the team has enough redundancy to remain operational without that single individual (Scaled Agile, 2020). This also means that to improve team efficiency the team is allowed time for retrospectives and feedback. Time spent on discussing improvements is not seen as waste but, instead as an investment to strengthen the team (Scaled Agile, 2020).

Table 10 describes the main dimensions (Thornton et al., 2012) of two ideal types of institutional logics within the product development institution.

Table 10. Ideal types of Institutional Logics in Product development

Dimension	**Resource efficiency logic**	**Flow efficiency logic**
Root Metaphor	Efficiency as utilization of man-hours	Efficiency as optimization of workflow
Sources of Legitimacy	Personal efficiency	Team efficiency
Sources of Authority	Management	Measurements
Sources of Identity	Clearly defined roles	Competence and flexibility
Basis of Attention	Resource availability	Workflow delays
Basis of Strategy	Utilize resources	Optimize flow, reduce time waste

Endorsing new institutional logics can be a powerful way to shake up established competitive positions in an industry (Durand, Szostak, Jourdan, & Thornton, 2013). One such example is when Google embraced an open-source logic strategy that contributed to making it one of the most innovative and successful players in the industry (Durand et al., 2013). In this case, Google used institutional logics as a strategic resource. For many organizations, flow efficiency logic has been implemented as a strategic choice (Tay, 2016) to replace resource efficiency logic.

Previous studies reveal that organizations do not necessarily need to reject old institutional logics while embracing new institutional logics (Durand et al., 2013). Again, Google is a good example of an organization that, while supporting open source initiatives, remained faithful to the

market logic of proprietary software with its search algorithm (Durand et al., 2013).

6.3 Synthesis and theoretical propositions

This section offers a set of theoretical propositions that will help guide the multiple case study. My theoretical argument in this chapter is that I have proposed *Agile ways of working* and *product development* as institutions. Each institution carries its own dominant set of logics that will act as the sense-making framework for employees working in large-scale ASD organizations. For Agile ways of working, I have proposed two institutional logics: *Agile toolbox logic* and *Agile rulebook logic*. These disparate views on implementing and tailoring ASD are well defined in the literature (e.g. Karlsson & Ågerfalk, 2009; Fitzgerald et al., 2006), but have not previously been conceptualized as institutional logics.

For the suggested institution of product development, I have also proposed two institutional logics: r*esource efficiency logic* and *flow efficiency logic*. These views on product development are also defined and described in previous research (e.g. Joglekar & Ford, 2005; Tay, 2016), but have also not been conceptualized as institutional logics.

Since none of the proposed institutional logics have been conceptualized previously, their dimensions (or *elemental categories*) have not been conceptualized either. All definitions in the different dimensions presented in this chapter (i.e., root metaphor, sources of legitimacy, etc.) are proposed based on previous studies of the conceptualized institutions (e.g. Karlsson & Ågerfalk, 2009; Fitzgerald et al., 2006; Joglekar & Ford, 2005; Tay, 2016). In the three case chapters 7, 8, and 9, the dimensions will be analyzed based on these theoretically proposed definitions.

Large-scale ASD organizations and their employees are exposed to their institutional contexts, but at the same time they can enact their agency to abide by one of the dominant institutional logics. These choices operate at the symbolic and material levels (Ferrer Conill, 2018) shaped by organizational history, culture, identity and goals (Thornton et al., 2012). Analyzing these can explain how employees in organizations adhere to dominating logics. The institutional logics perspective provides me with the theoretical tools to analyze and conceptualize why different tailoring decisions were made. This, in turn, could help me explain different impacts as the result of differing tailoring decisions.

7 Case Auto

In this chapter, the goal is to highlight relevant findings that are particular to this case. The first three sections (7.1, 7.2, and 7.3) present an answer to RQ1 (*How are inter-team coordination routines performed and tailored in large-scale Agile software development?*) in this specific case. In these first three sections, the case study is presented as a narrative with a structure similar to a play or movie, with three-acts; A First act (the setup or exposition); a Second act (containing rising action and further insights); and a Third act (which features the resolution of the story, in this case, my final observations). Each underlying subsection is named based on the themes created through the qualitative analysis of the studied implementation of routines. Hence, the underlying subsections will differ between the different case chapters presented in this thesis.

The narratives detail how the coordination routines are performed. This is to give a thorough understanding not only of the performed routine, but also to offer insights about values, priorities, and decisions, i.e. the institutional logics at play.

Therefore, after the narrative, an institutional logics analysis is performed and discussed in section 7.4 to answer RQ2 (*Why are inter-team coordination routines tailored (or not tailored)?*) for this case. This is followed by an analysis of coordination routines and dependencies in section 7.5 to provide further answers to the research question.

The perceived impacts of the implemented and tailored routines are analyzed and presented in section 7.6 to present an answer to RQ3 (*What perceived impacts (benefits and drawbacks) are associated with the implementation of inter-team coordination routines?*) for this case.

Finally, in section 7.7, perceived changes to autonomy are presented in order to answer RQ4 (*How is team autonomy perceived to be changed after implementing inter-team coordination routines?*) in this specific case.

The headquarters of the organization, as well as the majority of employees at the Auto department, are situated in Sweden. One additional European site and two sites in America are also part of the department; hence all meetings and documentation are in English. Some of the interviews were performed in Swedish since it was the Swedish site that was visited and observed. This means that some quotes and texts presented in this section have been translated into English. All the names of employees at Auto have been anonymized, and instead of family names, abbreviations of their roles are used. For example, the anonymized name of the

Department Manager Esmond is *Esmond Mgr* in the text. The same goes for Release Train Engineers (RTE), Product Owners (PO), Scrum Masters (SM), and Developers (Dev). Table 11 presents the interview respondents and employees quoted at Auto.

Table 11. Quoted employees at Auto.

Alias in quotes	**Role at Auto**	**Interview date**
Tod RTE	Release Train Engineer	2017-09-14
Martin Dev	Hardware Developer	2017-09-27
Conrad PO	Product Owner	2017-09-27
Oswald RTE	Release Train Engineer	2017-10-11
Lisa SM	Scrum Master	2017-10-19
Eve Dev	Software Developer	2017-10-19
Susan Dev	Software Developer	2017-10-19
Marcus SM	Scrum Master	2017-10-24
John SMPO	Scrum Master and Product Owner	2017-10-25
Esmond Mgr	Department Manager	2017-11-01
Nicole Dev	Software Developer	2017-11-02
Baptiste Dev	Hardware Developer	2017-11-04
Jonny Dev	Hardware Developer	2017-11-08
Andrew Dev	Software Developer	2017-11-08

Auto (pseudonym) is a software and hardware product development department at an automotive company. The studied department introduced Agile ways of working in self-organizing teams in 2013 and started implementing inter-team coordination routines in January 2017. Before implementing these routines, inter-team coordination was performed in an unstructured way mostly based on ad-hoc meetings.

Auto is organized into three Agile Release Trains with one Scrum Master and mostly one Product Owner per team (a few Product Owners manage two teams). Each team is organized locally, at one site, but each train consists of teams from all sites, with a majority of teams in Sweden where the headquarters of the organization resides. Three people have the role of Release Train Engineer (who coordinate each of the three trains), and one person, Esmond Mgr, is the manager of the department.

According to Esmond Mgr, there were two reasons for implementing inter-team coordination routines at Auto. In 2016, before the implementation, Auto was an organization that was overloaded with work. People were stressed and worked too much overtime, and there were signs that people were not feeling well. The other problem was regarding estimates and commitments. Sometimes the department was asked to perform development work, which was estimated for thousands of hours, and the

management team asked themselves: "How do we know that we will be able to deliver what we take on?".

Esmond Mgr, and the rest of the management team, looked into routines suggested by the framework SAFe and believed that implementing some of them would solve, or at least help them with, both problems.

Conrad PO, one of the Product Owners who were part of implementing the routines from the start, addressed a third reason. He explained that Auto had seen great improvements in every single team due to the implementation of Agile ways of working. The decision to form teams locally on every single site had made the teams autonomous and efficient. In seeing these improvements, managers realized that the organization needed improvements on a team-of-teams level, both regarding coordination of solutions to emerging dependency issues.

Tod RTE expressed yet another reason for the implementation. Managers at Auto sensed a lack of personal, overall responsibility for the product. Tod reflected that it is probably a natural development when organizations grow, but he also thought that the focus on individual commitments in the autonomous team could have led to a lacking responsibility for the overall picture. By implementing routines where teams interact more, Auto thought this development could be stopped.

Oswald RTE addressed that the inter-team coordination routines were implemented from the ground up. According to Oswald, a better approach would have been to start with middle-management.

> Where I have read [about implementing inter-team coordination routines], they are advocating that you … do not start at the team level. You do not start at [the highest] portfolio level. Instead, you start at the program level. But we have definitely implemented from the bottom up. If we had started in the middle, and made those who lead feel confident in how we work with Agile, then that would have been reflected down on those who ... it would have been much easier to implement on team level after that. (Oswald RTE)

Oswald further explained that although Auto received help from an Agile coach when implementing Agile ways of working from 2013 to 2015, they did not take help when inter-team coordination routines were implemented. Instead, managers decided to implement routines suggested by SAFe, but picking and choosing routines based on the needs at Auto.

According to Marcus SM, one reason for implementing routines suggested by a framework such as SAFe was that it was easier to argue for this when the company realized that their competitors were becoming

more efficient. In such times, Marcus SM argued, it is easier to rely on proven methods than to invent methods on your own:

> [The company] has started to gain some bad confidence because of competitors who catch up and sometimes outmatch us … it has become easier to argue that things should be done properly and according to methods recommended by experts. (Marcus SM)

Figure 9 shows a timeline of the performed observations at Auto.

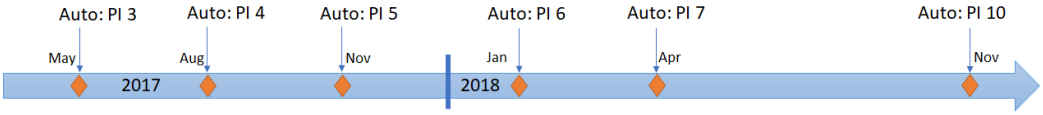

Figure 9. Timeline of PI planning events when data was collected at Auto.

Below, the first act presents the planning and execution of PI 3, the second act PI 6, and the third act PI 10. The reasons for not reporting every observed PI planning workshop are several. One reason is the amount of data collected. It might become too repetitive to read how the routine was performed each time with small differences between each workshop. The differences are easier to present by explaining how routines were performed and changed with longer intervals. Another reason is that the exact date and time for a change is not relevant for the study. It is even difficult to mark the exact time of some decisions since they often started as small experiments and evolved to become noticeable tailoring of the routine.

Auto had already started its implementation of SAFe when I first visited them in the spring of 2017. At that time, they had already reached their third PI, hence the first act starts in the PI planning days of PI 3.

According to the Oswald RTE, the two initial PI planning events were not very successful.

> We … [thought that once] we get PI planning in place, we can appoint roles like Release Train Engineer and Product Manager … later. The result was that confused project managers had to take these roles who had not received [SAFe] training. So in hindsight, we should have focused more on training and established new roles before we started with [the first] PI planning. (Oswald RTE)

But according to Marcus SM, this confusion was not perceived by the teams.

> It felt like it was very well thought through when it was launched at [Auto]. What was most difficult was probably what our teams were expected to do

> by ourselves during PI planning and understand how to handle stretch objectives in a good way. (Marcus SM)

Tod RTE further confirmed that this possible confusion during the first PI planning events did not cause any problems to the organization.

> It wasn't like we started [to work according to SAFe] and then had two months when everything was 'up in the air' when nothing really gets done. Instead, all the daily operations worked fine. (Tod RTE)

7.1 The First Act: PI planning for PI 3

On Tuesday May 30, 2017, planning was initiated at one of the three Agile Release Trains. All three trains were planned sequentially to allow managers and stakeholders to attend all of them. A very detailed planning agenda was presented, depicted in Figure 10.

PI planning

Tuesday

Start	Stop	Duration	Topic
08:30	09:00	00:30	Welcome day 1
09:00	10:00	01:00	Business context
10:00	10:15	00:15	Break
10:15	10:35	00:20	Runway update
10:35	11:35	01:00	Features in focus
11:35	11:40	00:05	Introduction team breakout
11:40	12:22	00:42	Lunch
12:22	14:35	02:13	Team breakout
14:35	14:50	00:15	Scrum of scrum's
14:50	15:05	00:15	Break
15:05	16:35	01:30	Team breakout
16:35	16:50	00:15	Scrum of scrum's

Wednesday

Start	Stop	Duration	Topic
08:00	08:10	00:10	Welcome day 2
08:10	09:00	00:50	Team breakout
09:00	09:15	00:15	Break
09:15	10:40	01:25	Team breakout
10:40	11:30	00:50	Final plan review

Figure 10. Planning agenda at Auto.

After a few minutes of initial greetings, each team had a few minutes to present what they had done in the previous PI. For *Business context*, one of the top managers in the organization presented important projects. He pointed out the importance of focusing on fuel economy and presented figures showing competitors' results in independent tests. He called out for individual initiatives to increase fuel economy and expressed that he

would prefer to see more developers' initiatives in backlogs rather than ordered features from program managers.

The *Runway update* consisted of information on important decisions from the three Communities of Practice. The first presenting Community of Practice focused on diagnostics, the second on architecture, and the third on verification.

During the agenda item *Features in focus*, project managers presented information from different projects. This was called *Project postcards* since all relevant information was gathered in one slide with an appearance similar to a postcard. Since many projects involved teams belonging to different Agile Release Trains, it was important for the project managers to adjust their postcards for each PI planning session with the three different trains. Also, this was a way of getting away from reporting status in every single project and instead force project managers to summarize information. Figure 11 shows an example of a Project postcard with confidential information blurred.

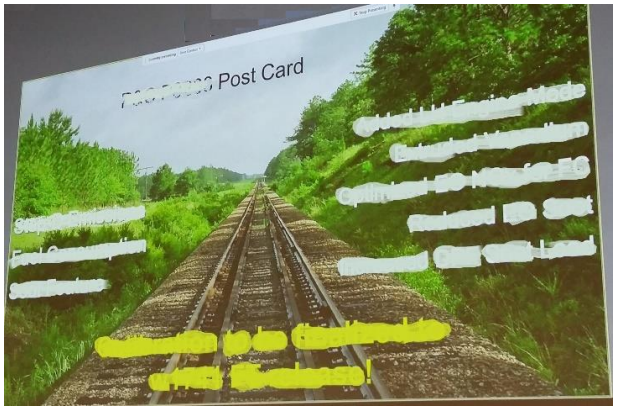

Figure 11. Example of Project postcard at Auto.

After almost each agenda item, posters were put up on walls to collect feedback on each part of the PI planning event (see Figure 12). The posters were divided into three parts. Two were free text areas to give comments about *What did you like* and *Change one thing*. The third area gave attendants a possibility to rank the event on a scale from one to five by applying a post-it note on the perceived level. On this post-it, the attendant was encouraged to write a comment to give further feedback.

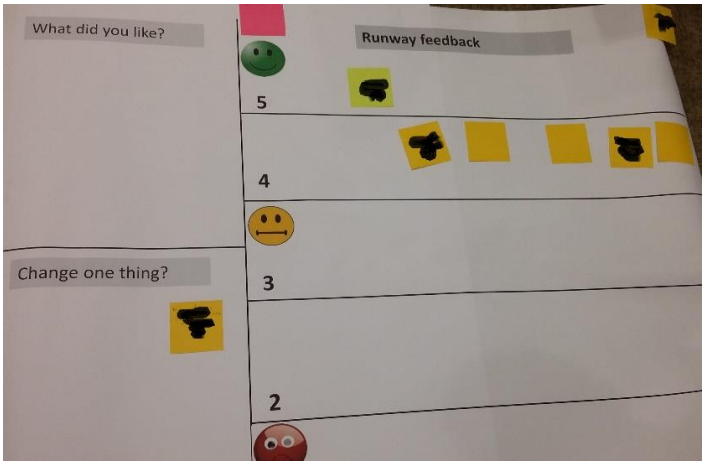

Figure 12. Example of feedback poster for Runway update at Auto.

During the *Introduction team breakout*, Oswald RTE displayed a picture to remind everyone of what the main output of PI planning should be: PI objectives (with a business value assigned), program board (showing dependencies), risk board, and team boards, i.e., plans for each team. The next slide showed a template called *PI objective sheet*, which should be used by each team to present in the final plan review. The template contained two lists: *PI objectives* and *Stretch objectives* with *Business value* for each objective. At the bottom of the template, the team average *Confidence vote* should be displayed by a number of *thumbs up*-icons.

As *Team breakout* started, all teams went into different meeting rooms to plan for the upcoming PI, consisting of three sprints with a duration of three weeks each. During the team breakout session, all Scrum Masters met for short *SoS check-in* meetings, which is a mini *Scrum of Scrums* meeting conducted during the PI planning event. The SoS check-ins (in the agenda: *Scrum of scrum's*) consisted of a virtual meeting with representatives from all teams attending. A slide presented during *Introduction to team breakout* reminded everyone that each team must send a representative to the meeting, but everyone is free to listen in. During each meeting, all Scrum Masters updated a status list, see Figure 13, with color-coded symbols.

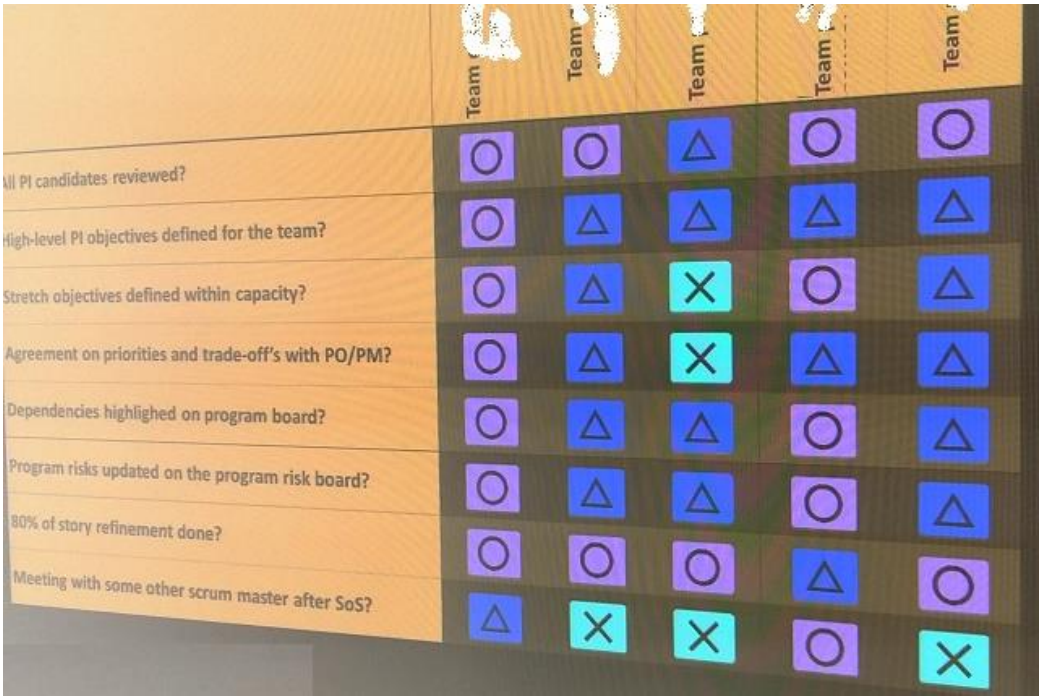

Figure 13. The status list used at the SoS check-in meeting at Auto.

A circle (in purple) meant that the question had been answered or the activity performed. A triangle (in blue) meant that it was started but not answered of finished. A cross (in turquoise) meant that it was not started. These symbols were updated during each meeting when a spokesperson for a team, often a Scrum Master, addressed the shift in status.

At the *final plan review* on the second day, all teams presented their objectives with the use of the introduced template. No changes in plans were called for.

This PI, as all previous PIs at Auto, was based on a fixed ten-week cadence. Besides the three sprints of three weeks each, a final Innovation and Planning (IP) sprint with a duration of one week concluded the PI. The IP sprint consisted of a half-day *Inspect and Adapt* meeting, one-and-a-half-day for the next PI planning event, and the remaining time of the week was used for *hack days*. Hack days at Auto meant time to innovate individually, in pairs, or in teams. Employees chose what to do individually, and whatever results they came up with, were presented in a demo on Friday afternoon, from one o'clock.

During the *Inspect and Adapt* meeting at the end of the PI, Oswald RTE concluded that 78.4 percent of the planned work had been completed in PI 3.

7.1.1 Few Scrum of Scrums, more like status meetings

During the first PIs, Auto only conducted one Scrum of Scrums meeting per sprint, i.e. three Scrum of Scrums for a whole PI. Tod RTE talked about the format and challenges of the Scrum of Scrums so far.

> We go through the status [of each team], dependencies, and risks. That's the only thing we're talking about. We've had some trouble getting the meetings to work, and we've had trouble getting the teams to ... getting them to be the team's meetings. They just say 'we have nothing to talk to the others about' but if you ask leading questions, things turn up. But it is difficult to get the teams themselves to get involved in this, to share between the teams. (Tod RTE)

A Product Owner, Conrad PO, confirms the problem of getting Scrum of Scrums meetings to work. He points out that they tend to be more like traditional status meetings when they instead should focus on proactivity and solving possible future problems.

> [The teams] easily get stuck in only telling you about everything they do. It becomes like pure reporting. We try to steer their focus towards talking about risks instead. I think the most important thing … is to nurture a mentality that as a team, you control your own risks, follow up on your own dependencies, that you are not passive, that you are proactive. Then you get a balance between when you need to ask for support versus what to solve as a team. It is a lot about coaching and parenting the team's mindset. (Conrad PO)

7.1.2 Three Communities of Practice as proposed by management

Tod RTE explained that there were three Communities of Practice at Auto at the time: Architecture, Diagnostics, and Verification. The Architecture community was the most well organized and had meetings once every second week arranged by a community leader together with a Core Team of four to five people. Tod RTE explained how it started:

> The architecture was falling apart altogether, and we did not know where to address those issues. So we helped [a Core Team] to start the first community. You could discuss whether the management should be the ones to start communities, but in any case, we did that. That's how we started the architecture community as an attempt to get the teams to talk to each other about architectural issues. (Tod RTE)

Tod further explained the status and challenges of the Communities of Practice at the moment.

> It doesn't work as we imagined! [laughter]. None of them work as we had imagined, but they have become different in strength, depending on how much space they have taken in the organization. The people [in the Core Teams] are people who are well known who run around and talk to people and so we have partly come to terms with solving the problems we had, but not at all in the way we had thought of. We still do not have all the teams involved. (Tod RTE)

According to Tod RTE, the two other Communities of Practice were so far not very organized. They had community leaders but few meetings and few members attending.

7.1.3 Unaccustomed to plan for longer periods of time

According to Marcus SM, the challenge during the first PIs had been to learn to plan for more than one sprint.

> Before SAFe, we had more flexibility to really change our planning completely after three weeks… earlier, if we discovered that we had missed something, then we just changed it when the sprint was over… So the slightly longer time horizon was what was difficult to get used to … For us, it really means that we have to have ... a better, clearer backlog that grows a little further into the future. We allowed ourselves to be sloppy in the past [Laughter]. (Marcus SM)

7.1.4 Semi-mandatory daily "fika"

Apart from the formal coordination routines at Auto, Oswald RTE also described an additional routine at Auto. The "fika"-tradition, the Swedish word for a coffee break, is common in Swedish workplaces. At Auto, each train had its lounge, and taking a coffee break in Oswald RTE's train between 9 and 9.15 was almost mandatory. And, as Oswald RTE explained, this tradition is not only about taking a break but also about being able to talk about work and solving issues informally: "[We have] semi-mandatory 'fika' every day in [the lounge] between 9 and 9.15 in order to be available to the teams".

7.2 The Second Act: PI planning for PI 6

Oswald RTE opened up the PI planning event of PI 6 on January 29, 2018 by presenting the same one-and-a-half-day standardized agenda (see Figure 10, above). Oswald RTE, who was previously also a team manager, was now a full-time Release Train Engineer. He expressed his confidence in his teams by stating:

> For me, as a coordinator, I just need to make sure you all know what to do, and already after four PIs, I felt that everyone knew what to do even before I told you what to do [laughter]. (Oswald RTE)

Oswald RTE also expressed, with pride, the results from the previous PI, PI 5: "Results from PI5 were roughly 90 % done and 10 % in progress, almost done!"

During the agenda item *Runway update*, Esmond Mgr presented the work conducted by the management team initiative at Auto called *WoW* (Way of Working). The team consisted of Esmond Mgr, the assistant Department Manager, and the three Release Train Engineers at Auto. One important area for *WoW* was to decrease the average backlog waiting time. The average waiting time for all previous PIs had spanned between 17 and 39 weeks. The goal was to reach below ten weeks on average from the decision of building a feature until it was finished.

When presenting *Introduction team breakout*, Oswald RTE reminded everyone of a decision to change the final plan review template, which all teams should use for their final presentation. Besides expressing PI objectives and stretch objectives with their estimated business value, there should be closing criteria presented for each objective. These closing criteria should be suggested by the teams, but Product Owners should act as coaches in finalizing definitions of these criteria.

When all the teams started planning during team breakouts, Oswald RTE discussed one of his experienced difficulties with Esmond Mgr:

> One of the things that are difficult for us is to get this overview of progress as we move forward during the day. If we had all been in the same room [and not distributed on different sites], we could have just slammed all the features on one wall… Then we could have seen how planning progressed physically, but we must try to find another way. (Oswald RTE)

Another thing Oswald RTE put forth in his discussion with Esmond Mgr was the difficulty in planning due to important tests. He expressed that, although the teams knew in their plans that prototypes with new software were going out in field tests, they did not allow time for correction of error reports in their plans. They knew that error reports always would turn up after tests, just not how many and how much work it would mean. Esmond Mgr and Oswald RTE decided on always suggesting a percentage of leeway in plans for the teams after each field test. The percentage would, however, differ from case to case, based on the type of field test.

During a break in the team breakout session, Oswald RTE reflected on the whole PI planning event. He claimed that the major importance of the PI planning session is to be *seen* if you are a manager or Project Manager. If you are not seen, people might stop prioritizing your demands and opinions. As Oswald RTE saw it, many managers or Project Managers made an effort to come to sprint demos for the same reason: to be seen and avoid the risk of negligence.

> You also get the chance to be seen… at least, many get the chance to be seen, not everyone. But then we have sprint demos instead where those who can't be seen at PI ... can be seen there instead. (Oswald RTE)

After team breakout sessions and presentations of the teams' plans, Marcus SM expressed that this was the first PI planning event that really went smoothly, according to him. He also said that SoS check-in meetings during team breakouts worked according to the intended purpose:

> The [PI] planning we did … was really the first one where I would say it felt like we really mastered it … where we did, in my opinion, right… It went pretty fast, and everyone had inputs to our Scrum of Scrums sync, and it feels like we got what we wanted from those meetings. It was not just that we said there was a dependency, and then nothing more happened to it. Overall, it feels like we have learned very well how we want to work with this. (Marcus SM)

Conrad PO expressed a similar opinion of the successful PI planning event and also talked about the team members. According to Conrad PO, there had been fears expressed of not being allowed to specialize because of the announced aim of having cross-functional teams. In his opinion, that fear had blown over: "Some were afraid that they would not be able to specialize, but over time I feel that that discussion has settled down". But there are, as Tod RTE expressed it, always areas of improvement.

> Where we have problems is when we have dependencies on the other trains. Within the train, it works relatively well ... It becomes a little more difficult when you talk to another train, and they like 'Aaah, but we have our plans here, you see' [laughter]. I would say that it goes quite well when we find dependencies within the train, but between the trains, we can be more efficient. (Tod RTE)

During the wrap-up of the PI planning event for PI 6, John SMPO, who was both Product Owner and Scrum Master of his team, expressed his view of the final *Inspect and Adapt* routine which would be held just before the next PI planning:

> Something that we would benefit from more is this inspect and adapt [routine], I don't think that we ... we work too little with that. Well, I feel like the whole basic idea of Agile is that we should learn. Think it will get better now when ... our team manager has become a full-time Release Train Engineer. Then he will have more time for it simply to prepare and think [about the Inspect and Adapt routine]. (John SMPO)

At the *Inspect and Adapt* meeting in the IP sprint of PI 6, Oswald RTE presented that 102 features out of the planned 114 features had been developed and implemented, i.e. 89 percent. These numbers were equivalent to the previous PI, PI 5, as Oswald RTE declared at the beginning of PI planning for PI 6.

7.2.1 More frequent Scrum of Scrums

From having Scrum of Scrums meeting once per sprint, Auto increased the frequency of the meetings to once per week.

> All teams have Scrum of Scrums at 08.45, … so they go there directly on Mondays [after each team's daily scrum]. We don't have it every day, but once a week. [We] try to keep the timebox: four minutes per team. (Oswald RTE)

During the four-minute timebox per team, the Scrum Masters announce whether there are any delays in their plans or whether anything will be delivered ahead of time. They also explain whether they have encountered any impediments or identified new risks or dependencies.

> We also plan to invite everyone who owns risk ... maybe not to every Scrum of Scrums but at some point per sprint to follow up [on the risk] … and when they no longer have a risk, they don't need to attend. (Oswald RTE)

Auto still sees areas for improvement on how Scrum of Scrums is conducted. Marcus SM is not satisfied with how they are conducted and presents his view of several problems.

> It doesn't seem as the Scrum Masters, including myself, are really doing their job in bringing valuable input, new dependencies, and risks to these meetings ... it becomes a bit like a status report for your own team that only touches your own team … It's like some kind of hen-party where you just say 'We have this problem. It doesn't concern you, but you may listen anyway'. I think we could have made those meetings better … by just having them at a different time than as the first thing on Monday morning. It does matter when you have it and how relevant it feels. (Marcus SM)

7.2.2 Communities of Practice taking up more space

When *Features in focus* were presented during PI planning, the three Communities of Practice informed about decisions and intended upcoming work in the PI. The community leader of the Architecture community expressed gratitude in allowing more work suggested by the Communities of Practice to be planned for in the PI. During a short break after this presentation, Tod RTE explained the background to allow more work suggested by the communities:

> [The Communities of Practice] may have become more operative than we had imagined. We thought they would be more involved in talking about design and solutions and information materials and learn from each other. The community leaders have almost become Product Owners with their own backlogs, which can almost compete with our backlogs. That is why we have … set a boundary between enablers [which they can decide on] and features [that we decide on] so that they do not get the upper hand. That's why we said 70/30 features versus enablers in this PI planning. But they want more mandate. (Tod RTE)

Marcus SM, the community leader of the Verification community, expresses how the community works right now.

> We meet almost every other week … Everyone who participates has a focus factor that they adjust to have some time to spend in this community. And then, typically, we identify things that we could improve on. What we have decided is that we should be actionable. Not only should we say 'this would have been nice to fix', but instead ... in the end, after the meeting, someone should be in charge of getting the thing fixed at a certain date. (Marcus SM)

7.2.3 Challenges in horizontal communication

Tod RTE expressed his view on the challenges in communication horizontally in the organization.

> [I didn't expect] how difficult it would be to get ... and still is ... to get everyone to communicate ... so to speak ... horizontally. At all levels. It has been more difficult than we actually thought. It's still one of the things we wrestle with. Even at the team level as well. We managed to get self-organizing teams but then to get the teams to start communicating with each other and to make some bigger plans and communicate without anyone being there to help them... it has been tough. (Tod RTE)

Esmond Mgr explained that, at some point, they understood that the problem with coordinating between the three trains at Auto needed to be prioritized. They decided to force representatives from each train to participate in the PI planning event for the two other trains. Therefore, team

breakout sessions needed to be planned so they did not collide, thereby allowing participation in all three planning sessions.

7.2.4 Low attention to the program board and risk management

The presented features during PI 6 were only the ones with dependencies on other features, not all features. They were also fewer than during previous PIs. Conrad PO claimed that the reason for this is that management had not followed up on dependencies on the program board during PIs. Also, they had not followed up on the risks announced at the end of each PI planning event, which should be managed on the risk board.

> Before [this PI], we have had many things on the program board and risk board, but there are fewer now. Could it be because we don't follow up on them [irony]? [laughter!]. We complain and complain that they HAVE to fill them in, and then no one talks about them or help them. (Conrad PO)

7.3 The Third Act: PI planning for PI 10

PI planning number 10 began at one o'clock on November 6, 2018 and was divided into three days instead of two. This was due to a logistical issue, as expressed by Oswald RTE.

> A big problem is the premises, actually. It's such a sad thing to say, but... especially since we are a large company … with quite a lot of premises. But if the whole company should start … this kind of PI planning it will get hard to get hold of meeting rooms. (Oswald RTE)

The agenda of the PI planning did not differ more than in date and time for different agenda items but was essentially the same as it had been since the first observation of PI 3. The above-expressed frustration on lack of premises was due to the need for postponing the final presentation to the morning of the third day instead since no rooms were available to fit the whole Agile Release Train at the end of the second day.

The agenda item *Business context* was reduced to only 30 minutes, compared to previous agendas where one hour was used for this part. According to Oswald RTE, the reason for cutting down on this was feedback from several employees. They had expressed that overall long-term information was more useful earlier when they were starting up the Agile Release Train. Now, much of the information seemed like waste of time, according to the employees. Hence the decision to reduce the amount of business context information.

After presentations of *Project postcards*, Conrad PO talked to John SMPO, who is both the Product Owner and Scrum Master of his team. Conrad PO expressed what he perceived to be the benefits of presenting Project postcards.

> I think it's great that the project managers come here and create an overall picture. In the past, you only heard what was important for the local site. What they were doing [here]. This has really improved transparency. (Conrad PO)

John SMPO responded by expressing his view of the benefits of this new kind of PI planning, now that everyone had become accustomed to it.

> You get a much better understanding of the work ahead of you ... all [PI] planning focuses on that, to visualize what you need to get done. You get a clear overview of what everyone is doing, and that is a huge advantage, I think. (John SMPO)

During the *Introduction team breakout*, Oswald RTE announced that they would no longer use the feedback posters. Feedback had continuously decreased as fewer post-it notes were put up. Oswald RTE thought that an explanation for this might be that many feedback requests had already been expressed, since the posters had been put up for many of the agenda items. Instead, Auto decided to use the software *Mentimeter* to ask for feedback throughout the PI planning event.

Another thing Oswald RTE addressed was that teams should always put up their planned features *before* each SoS check-in during the PI planning event. In order to be able to discuss and solve dependencies, Oswald RTE explained, features need to be visualized on the program board.

Compared to the previous PIs, when all PI objectives and stretch objectives should be given a business value, Oswald RTE explained that this would not be needed during this PI planning. Since these business values were not used for anything, neither prioritization nor follow up, Auto had now decided to drop the use of them. Instead, as Oswald RTE pointed out, it was important to be clear and precise when naming objectives so everyone could understand the value of them.

As always, Oswald RTE presented the number of planned versus completed features at the *Inspect and Adapt* meeting in the IP sprint of PI 10. Oswald RTE proudly announced that all 104 planned features had been developed and implemented, i.e. 100 percent. This announcement was greeted with applause and whistles.

7.3.1 Project managers bridging coordination with other departments

Within Auto, the three trains synched PI planning so that team breakout sessions did not collide. In that way, representatives and stakeholders could attend all three PI planning events to coordinate dependencies between trains. There was, however, also a need for coordination with teams in other parts of the organization, outside Auto.

John SMPO thought that Auto did not really have a way of dealing with dependencies outside their own department except using the Project Manager to coordinate the work.

> We discussed that in the PI [retrospective], that we really have… we have no good way to work with [coordination between trains] yet. So we involve the Project Manager at [department X] more. So he has been with us and looked at our PI planning, and then they have their planning next week. So they will know what we will work with at least. (John SMPO)

7.3.2 Using longer plans for coordination, shorter for flexibility

With a PI length of ten weeks, the teams at Auto had been forced to plan further into the future than they were used to. According to Marcus SM, this had helped the teams foresee coordination needs. At the same time, by keeping the sprint length to three weeks, just as before the introduction of inter-team coordination routines, flexibility to change priorities was not lost.

> I'm starting to get very satisfied with the PI planning, because ... well, mainly because it has had a feedback effect on us as a team. We have become very good at predicting. One could argue that it may not be so Agile to think too far into the future, but when it comes to dependencies … it is still quite important. And that's where we've become really, really good. It's quite nice that you still have the opportunity to be Agile between every sprint. At the same time as you have identified dependencies as much as possible in the slightly longer term. (Marcus SM)

John SMPO confirmed this view from the experience of his team. Long-term planning with short sprints still allowed for flexibility.

> At first, we thought it was difficult to plan so far ahead. But as of late, we have learned that this plan can change over time, so now ... we have three week sprints, and then we can re-evaluate and re-plan after three weeks. (John SMPO)

7.3.3 Preparations not coordinated enough

Although PI planning focuses on upcoming work, preparations are needed to have a prioritized backlog. Besides prioritizing a backlog, Auto claimed that there was a risk of shortsightedness if this preparatory work was not well coordinated.

> When you divide the work in this way, and you try to run short sprints and ... prioritize, there is a tendency that you do far too many small features and too little research ... or preparations like pre-study work. It is important not to … only solve what is closest to you and the urgent problems and forget about the long-term. Maybe this is not because of Agile, but I think it is amplified a little by the Agile way to work because … you are rewarded for completing tasks. (Oswald RTE)

What Oswald RTE suggests is that there is a risk of rewarding small gains in Agile ways of working.

7.4 Identified institutional logics

The decisions made, roles put into place, and collective actions conducted when implementing inter-team coordination routines shows the assumptions, values, and beliefs at Auto. I have investigated how they connect to the dimensions (or *elemental categories*) of institutional orders and thereby point to the dominant logic in each institutional order (Agile toolbox logic versus Agile rulebook logic, and resource efficiency logic versus flow efficiency logic). By combining these two dominant logics, I can present a model close to an institutional logic ideal type (Ferrer Conill, 2018) at Auto.

However, it is important to remember that this is a case study of an organization based on a certain time, place, and the current individuals attending. Categorizing an institutional logic ideal type does not mean that a case organization is always anchored in that type (Ferrer Conill, 2018). Adherence to logics and ideal types is fluid and changes over time. The changes may be due to changes in the world, such as a worldwide pandemic, or a result of a strategic decision (Durand et al., 2013). Table 12 summarizes the different dimensions at Auto that present the beliefs and values of specific institutional logics, creating an institutional logic ideal type consisting of Agile toolbox logic and flow efficiency logic.

Table 12. Institutional logic ideal type at Auto

Dimension	Agile toolbox logic	Flow efficiency logic
Root Metaphor	Agile routines as tools	Efficiency as optimization of workflow
Sources of Legitimacy	Unity of improvement needs	Team efficiency
Sources of Authority	Commitment to team decisions	Measurements
Sources of Identity	Attentiveness and responsiveness	Competence and flexibility
Basis of Attention	Routine inefficiencies	Workflow delays
Basis of Strategy	Invent and adapt routines	Optimize flow, reduce time waste

There is evidence of an Agile toolbox logic residing at Auto. First, Oswald RTE explicitly stated that the managers at Auto decided to implement routines suggested by SAFe but picking and choosing routines based on their needs. That is why they did not ask consultants specializing in SAFe implementations for help. Instead, they implemented chosen routines on their own.

Another evidence is decisions to stop conducting parts of routines suggested by SAFe. One such decision was to stop appointing business value to objectives. Since these business values were not used for anything, neither prioritization nor following up, Auto decided to drop them. Instead, as Oswald RTE pointed out, it was important to be clear and precise when naming objectives so everyone could understand the value of them. An example of when Auto invented a routine on their own was when project information was presented based on *Project postcards*. This was to get away from reporting status in every single project and instead force Project Managers to summarize information.

There is also one example where both evidence of Agile toolbox logic and flow efficiency logic is present at Auto. The *Business context* part of PI planning was reduced to only 30 minutes, since several employees felt that receiving this information often was a waste of time.

Esmond Mgr expressed that one reason for implementing routines from SAFe was that Auto was an organization overloaded with work. People were stressed, worked too much overtime, and had no leeway for improvements or innovation. Thus, the intention of this strategy was to focus on a flow efficiency logic based on SAFe values and routines.

One example of flow efficiency logic was presented during PI planning for PI 6, where Esmond Mgr explained the work conducted by the

management team initiative called *WoW*. The aim was to decrease the average backlog waiting time since the average waiting time in the previous PIs had spanned between 17 and 39 weeks.

Regarding an organization overloaded with work, Esmond Mgr and Oswald RTE decided on always suggesting a percentage of leeway in plans for the teams after each field test – a typical example of flow efficiency logic.

Allowing time for retrospectives, feedback, and innovation are all signs of flow efficiency logic, and SAFe suggests an IP (Innovation & Planning) iteration at the end of each PI (Scaled Agile, 2020). As previously mentioned, IP was performed in one week and consisted of a half-day Inspect and Adapt meeting, one-and-a-half-day for the next PI planning event, and the remaining time of the week was used for hack days. Hack days at Auto meant time to innovate individually, in pairs, or in teams and present results on Friday afternoon.

7.5 Analysis of coordination routines

The first version of this analysis was presented in Gustavsson (2019b). This section presents a more thorough analysis and further findings.

One question this analysis aims to answer is: does the way routines are carried out (the *how we do it*) differ from the prescribed way of performing inter-team coordination routines according to SAFe? This tailoring is the difference between prescribed and performed routines. Another question this analysis aims to answer is the reasons for tailoring routines (the *why we do it*). Table 13 summarizes the differences between how routines are prescribed by SAFe and how routines were performed at Auto.

Table 13. Routines at Auto from a tailoring perspective

Routine	**According to SAFe**	**Auto**
PI planning	Set cadence, PI plan every 8-12 weeks.	Set cadence, PI plan every 10 weeks.
	4*2 weeks sprints and IP sprint (2 weeks).	3*3 weeks sprints and IP sprint (1 week).
	Duration: 2 days.	Duration: 1,5 days.
Scrum of Scrums	Twice weekly.	Varied from once a week to once per sprint.
	Duration: 30 minutes and "meet after".	Duration: 20-30 min, no "meet after".
	Attendees: RTE, SMs, and possibly stakeholders.	RTE and SMs.
Program board	Contains feature delivery dates, feature dependencies, and milestones.	Contains feature dependencies and milestones, not all features.
	May, or may not, be maintained during PI.	Not maintained during PI.

Table 14 shows more detailed information about the PI planning routine at Auto, with duration and intervals of PI planning and PI periods.

Table 14. PI planning data from Auto at PI 3, PI 6, and PI 10.

	Auto PI 3	Auto PI 6	Auto PI 10
Number of weeks per PI	10	10	10
Number of sprints per PI	3	3	3
Sprint length (weeks)	3	3	3
Length of final Inspect and adapt meeting	None	None	None
PI planning workshop (hours)	12	11.5	11.5
Team breakout time (hours)	5.5	6.5	7.5
Team break-out time (percent):	45.8%	59.1%	65.2%

To summarize the tailoring conducted at Auto relative to prescriptions from SAFe (Scaled Agile, 2020):

PI Planning

- An increasing amount of team breakout time each PI
- Conducted virtually, teams in different places

Scrum of Scrums

- Varied from once a week to once per sprint
- Scrum Masters mainly reported progress
- Conducted virtually, teams in different places

Use and management of the program board

- Used a digital board since teams where located in different places

The reason for having a strict cadence was explained in the same way by Tod RTE and Oswald RTE: by having a strict rhythm, team members were better at predicting and estimating, and that helped Auto to be more precise in predictions towards stakeholders. Auto thought that one-and-a-half-days would probably be enough time for the teams to finalize their plans and decided to conduct the workshop using a little less time for each item than prescribed in the standard PI planning agenda of two days (Scaled Agile, 2020). During my final on-site visit, when Auto held their tenth PI planning workshop, the overall format was still the same, but the detailed contents of the PI planning workshop had been tailored all along, based on feedback from people at Auto. Notably, the amount of time spent on presentations compared to team breakout time changed over time. In PI planning of PI 3, 45.8 percent of the time was spent on team breakout planning, but this changed after criticism from employees about them spending too much time on presentations. During the last studied PI planning workshop, PI 10, 65.2 percent of the time was spent on team breakout planning. Instead, the formal presentations and meetings during PI planning were shortened and tailored along the way.

Auto acknowledges that Scrum of Scrums had not been prioritized in their implementation of inter-team coordination routines. From having the meetings only once per sprint, they increased the cadence to once per week, which is still less than prescribed by SAFe (Scaled Agile, 2019). No matter how often Scrum of Scrums meetings were conducted, they were performed in the same way, where one Scrum Master at a time addressed what they had done since the last meeting and discussed whether they had problems or projected problems, i.e. risks. Release Train Engineers viewed the meetings as an arena for highlighting risks, while Scrum Masters saw it mainly as a place to report progress. None regarded the meeting as a place specifically to resolve dependency issues.

Auto decided from the very start only to use the program board to highlight features with dependencies between teams and other departments, not to show all features. Significant milestones were also displayed. This was observed in the same way during every on-site visit. Instead of using a physical board, Auto used the online tool iObeya[5] as a program board. The program board was only updated during the PI planning and was not kept up to date during the PIs during the observed on-site visits.

[5] https://www.iobeya.com

However, the importance of having the program board up to date during PI planning was highlighted during PI 10, making it the central planning artifact during PI planning.

7.5.1 Dependency analysis

In this section, dependencies are analyzed from three different perspectives. First, an analysis based on the coordination theory (Malone & Crowston, 1994). Tod RTE gives a technical description of the feature dependencies at Auto.

> Many of our features have dependencies … due to our software [platform] architecture. Most of the dependencies … are between our software platform and our functionality, where the software platform is almost like an operating system to the hardware. Whenever you start to change overarching functionality, they [the people working with the software platform] have to get involved, and you have to start synching with them. (Tod RTE)

Tod's statement is in agreement with other respondents, who express a typical *producer-consumer dependency* issue. Two sub-dependencies seem important: *transfer*, ensuring that the output is available to the consumer when needed, and *precedence,* making sure the actor performing the second task learns when the resource is available, and the task can be started.

Based on Thompson's (1967) classification of terms of workflow between actors, a second analysis can be performed. From interviews, it seems like most dependencies could be described as *sequential dependencies* or *pooled interdependencies*. Pooled interdependencies are often referred to as competence or knowledge dependencies by the respondents. The view on which of them was in majority varied between respondents and between teams. John SMPO worked in a team that focused on testing. He claimed that for them it was all about sequential dependencies: "We are most dependent on others to get something done so that we can test it."

Lisa SM saw both kinds of dependencies, but sequential dependencies were more common: "About 2/3 sequence [dependencies] and 1/3 competence [dependencies], I would say."

But according to Marcus SM and the hardware developer Baptiste Dev, there was more pooled interdependencies than sequence dependencies in their team.

> Usually, it is some kind of knowledge dependency, I would say. Yes. It has happened that we have had [sequential] dependencies where we expect something else to be done before we can continue. But it's not common. It's mainly knowledge dependence. (Marcus SM)

> We don't have many hard dependencies, so I would say that almost everything is competence [dependencies]. We understand better how to build stuff on our own, depending on the competence of other teams. (Baptiste Dev)

To summarize, both sequential dependencies and pooled interdependencies are common at Auto and vary between teams.

The third important area to analyze regarding dependencies is static versus emerging dependency issues, as emerging dependencies are claimed to be more important than has previously been put forth in research (Okhuysen & Bechky, 2009). Five respondents expressed their view on how common emerging dependencies were at Auto.

> I would say maybe not more than 25 percent [emerging dependencies], maybe even less. (Baptiste Dev)

> My feeling is that of the features we actually have in the lists in the PI planning, I would say that a fairly satisfactory majority of them do not appear as new dependencies. I would say up to 80 percent of the dependencies are identified. (Tod RTE)

> You can come up with new risks [during the PI], but not any new dependencies... you have pretty good knowledge about that during [PI] planning. (John SMPO)

> Not much, I would say ... stuff turns up, and they could possibly be dependencies. But [compared to what] we have planned, I do not feel that we discover many new dependencies. (Oswald RTE)

> So far, I have not experienced that many new [dependencies] are emerging at all. (Marcus SM)

Although these are individual voices from only five respondents, they suggest that emerging dependencies are not very common, and not a big problem, at Auto.

7.6 Perceived impacts

At the beginning of the planning event for PI 6, a paper questionnaire was handed out to all participants to be filled out by the employees attending PI planning. The full questionnaire can be seen in Appendix B.

Several parts of the questionnaire were inspired by Laanti et al. (2011), and the first section was intended to give an understanding of the employees' view of the different parts of SAFe. Statements with Likert scales where used, ranging from 1 to 5, where 3.0 can be considered neutral. The first section of the questionnaire investigated the employees' view on the implemented inter-team coordination routines and is the only section

used in this analysis. The results of the survey are presented in Table 15. At Auto, 125 employees out of 141 answered the questionnaire (a response rate of 88.7 percent).

Table 15. Perceived impacts of inter-team coordination routines at Auto

Survey Question (Likert scale 1-5: 1=Not true, 5=Very true)	**Mean**	**SD**	**Median**	**Mode**
The PI planning gives me a good overview of our work	4.2	0.7	4	4
The Program Board is very helpful for coordination between teams	3.4	1.0	3	3
The Scrum of Scrums (SoS) meetings solve inter-team coordination problems	3.3	1.1	4	4
There are many problems with working in a large-scale setting as ours, using SAFe	3.3	1.0	3	3
I would like to go back to the old way of working (instead of working according to SAFe)	1.9	1.1	2	1

As Table 15 shows, the respondent's opinion on the PI planning event gave a Likert-scale average of 4.2, which shows a positive view of its usefulness. The view of Scrum of Scrums averaged at 3.3 and the program board at 3.4, which shows a mildly positive view of these parts as well. The fourth statement proposed: *There are many problems with working in a large-scale setting like ours, using SAFe* averaged at 3.4, which is somewhat above neutral. The final statement (*I would like to go back to the old way of working (instead of working according to SAFe)*), with an average of 1.9 (and a mode of 1), shows that most employees would not like to go back and are positive towards the newly implemented inter-team coordination routines.

7.6.1 Perceived benefits

The first step of the analysis consisted of an investigation of all open-ended answers from Auto to familiarize myself with the data (Braun & Clarke, 2006). The second step was to generate initial codes by searching for patterns and grouping together answers, and the following steps resulted in a definition of themes. The themes can be seen in the right-hand column in the thematic map displayed in Table 16. The order of themes displayed in the table is based on the number of coded answers from the open-ended questions relating to each theme. This means that the theme displayed at the top of the table has the highest amount of coded answers from open-ended questions, and the following themes are displayed in falling order of the number of coded answers.

Table 16. Codes and themes of perceived benefits at Auto.

Codes from open-ended questions	**Codes from observations and interviews**	**Themes**
Visibility/Transparency [11]	Better overview [5]	*Overview and transparency*
Overview [9]	Increased transparency [4]	
Detection of impediments [1]	Transparency for managers [4]	
Coordination, synchronization, and collaboration [11]	Improved teamwork [5]	*Coordination and cooperation*
Structure [6]	Easier to help other teams [4]	
Better planning [11]	Precision in promising what do deliver [5]	*Planning precision*
Efficiency [5]	Faster testing possibilities [3]	*Speed, focus, and efficiency*
Focused work [2]	Less waiting time [3]	
Speed [1]	PI planning more focused planning [3]	
	More streamlined [3]	
Better prioritizations [6]	Flexibility despite long term plans [3]	*Clear priorities*
Motivation [3]	Learning from each other [4]	*Employee motivation*
Commitment/Involvement [2]	Employees express motivation [2]	
Even workload [5]	Less stressful [2]	*Stress relief*
	No negative effects despite transformation [3]	*No interference on teamwork*

The themes created for perceived benefits are explained below. Numbers within the brackets show the number of answers from the open-ended questions, as well as observations and interviews that belong to the same code.

Overview and transparency

The two most common answers to the open-ended questions were versions of "improved visibility/transparency" [11] and versions of "better overview" [9]. One answer, also relating to transparency, was the "possibility for easier detection of impediments" [1].

In interviews, three respondents talked about increased transparency [4], such as Martin Dev explaining: "PI planning makes everything very transparent". Respondents also found benefits of a better overview [5] where Marcus SM explained how this had evolved in the team:

> PI planning ... has had a feedback effect on us as a team. We have become very good at predicting. One could argue that it may not be so Agile to think far into the future, but when it comes to dependencies … it is still quite important to get a better overview. (Marcus SM)

The improved overview of work was also mentioned several times during observations.

In interviews, Developers suggested that the new way of working meant a lot more transparency for managers [4]. Andrew Dev said: "I also hear from the managers that this has meant a lot to them. That they get much better insight into what the different teams are doing and know what to expect".

Coordination and cooperation

The most common answers to the open-ended questions were variants of "better coordination/synchronization/collaboration" [11]. Several answers also suggested a "better structure" [6], related to the structure of cooperation.

In interviews, respondents said it was now easier to help other teams [4] thanks to the implemented inter-team coordination routines. For example, Andrew Dev explained: "Before, you had a poor understanding of what the other teams were doing, so you didn't know that you could easily have helped them". Susan Dev plainly expressed that "it's easier to help each other across team boundaries".

Also, respondents mentioned improved teamwork [5] due to better cooperation. A quote from one of the Software Developers gives a good example of the improved teamwork:

> [I] don't really know what picture I have given you about how it works, but I think it is better now than before we started with SAFe. Especially in the team. The team works better now than before… I remember that I thought a lot about that and yes, it really is better than before, in every way. (Nicole Dev)

Besides team members, Conrad PO explained his view that "Especially for team focus it has been very helpful … it has been a cultural journey of more collaboration, working together in the teams, and so on".

Planning precision

Many of the answers to the open-ended questions simply stated versions of perceived "better planning" [11]. In interviews, respondents talked about increased precision in promising what to deliver. They sensed this change as an effect of the implemented PI planning routine. One re-

spondent further developed this by expressing that he thought that the teams gain an overall picture due to longer plans, thereby increasing precision in the plans. As presented in the three Acts above from observations, planned work versus achieved results increased along the way. From 78.4 percent planned work completed in PI 3 to 89 percent in PI 6 and finally 100 percent in PI 10 for the Agile Release Train governed by Oswald RTE.

In several interviews, respondents mentioned improved precision in promising what to deliver [5]. The department manager expressed the benefit in the following way:

> One effect that has also come out [of the implemented inter-team coordination routines] that I am quite happy with, is that it has become a little clearer, it becomes a little easier for us to actually communicate back what we can and cannot do. (Esmond Mgr)

This was also confirmed in the interview with Nicole Dev, who said: "You could say it's much more predictable now than it's been before".

Speed, focus, and efficiency

From the open-ended questions, only one answer mentioned the word "speed" [1]. However, in observations and interviews, several respondents talked about increased speed both related to planning and work pace. Some mentioned the decreased waiting time between tasks with dependencies, which meant more focused work [2] on each feature. Others suggested that planning, with the PI planning routine, resulted in more focused planning.

Also, different versions of "efficiency" [5] were found in the answers. In observations of final plan presentations during PI planning, several employees expressed that the work seemed more streamlined. They explained the word streamlined as more efficient work where priorities were clear.

Respondents acknowledged that the PI planning routine meant more focused planning [3]. According to Conrad PO, people might think a lot of time is wasted on too many meetings due to coordination routines, such as the PI planning routine, which could lead to a lack of efficiency. However, Conrad PO did not perceive that efficiency had decreased but rather that it improved focus.

> There are some employees who think that because we spend more time on coordinated planning, there is a lot of time wasted. From my perspective, I

> don't think efficiency has gone down. But that's just my feeling; I have no quantitative measures to support it. (Conrad PO)

Another identified benefit was that there are now faster testing possibilities [3]. Some might think that testing would take longer due to more teams needing to cooperate in testing full functionality, but Andrew Dev stated that "on the contrary, testing has accelerated well and truly because we work like this". Yet another benefit expressed by respondents was that there was less waiting time [3]. Jonny Dev actually saw this as the major benefit of the new way of working: "Above all, we are faster in finishing work now. Before we had to wait a lot, but now, with PI planning and deciding things in advance, it's much faster".

Clear priorities

Different variants of "better prioritizations" [6], were described in the answers to the open-ended questions.

Respondents in interviews also talked about how priorities can be changed in every new sprint, not just during PI planning. This means that there is flexibility despite long-term plans [3]. A Scrum Master put it this way:

> My opinion is that many people in the team think that [PI planning] is a little different and that you ... you feel that you are planning very long-term, which you do to some extent, but you also plan for the possibility of plans to change. (Marcus SM)

This was also acknowledged in another interview:

> In the beginning, we thought it was difficult to plan so far ahead. But as of late, you have somehow learned that this plan is changing. So it is more that you plan for three week increments ... [even if] you 'set' ten weeks. (John SMPO)

Another benefit related to clear priorities was that respondents felt that work became more streamlined [3]. During a short unintended break in a PI planning event due to technical problems, a manager took the stage and asked for some general feedback. One person in the crowd answered: "We're much more streamlined", with many confirmatory nods and hums from the rest of the attendees. Andrew Dev also suggested this in an interview: "All planning becomes clearer and more, what do you say, streamlined. We all work in the same way now, sort of".

Employee motivation

Several answers to the open-ended questions expressed "increased motivation" [3] in their overall work. Some answers expressed in more detail that they felt "more commitment/involvement" [2] with the added routines.

Previously, I presented the perceived benefit of being able to help other teams, and interviews also showed another benefit in employees learning from each other [4]. The hardware developer Martin Dev expressed that "we learn more from each other because we talk to the software team more than we did before. And some teams are merged with both hardware and software [developers]".

Two respondents described that employees expressed motivation [2] due to the implemented changes. Esmond Mgr said: "Many people came personally and thanked us for making this change, that's how happy they were … I believe that most people feel that they are motivated by seeing a whole".

Stress relief

Variants of "even workload" [5] were expressed in the answers to the open-ended questions. One of the reasons for implementing the inter-team coordination routines at Auto was that it was an organization that was overloaded with work. People were stressed and worked too much overtime, and there were signs that people were not feeling well. According to Esmond Mgr, one intended impact of this implementation had been reached.

> You can very quickly see when people don't feel good at work with stress and such things, and much of it was actually dampened, and stress decreased significantly. (Esmond Mgr)

In observations, employees also expressed that they perceived the working climate to be less stressful [2] now.

No interference on teamwork

The data that formed this code did not originate from any of the open-ended questions. Instead, both survey respondents and voices during observations expressed that they did not perceive any negative effects on the teams despite implementing new routines [3]. They had in fact anticipated some disturbance, as their previous experience from change projects was that several areas were negatively affected before benefits were perceived. That was not the case during this transformation, according to Tod RTE who expressed that "all the daily operations worked fine" during the first

two months of implementation. According to the answers from three respondents, teams worked efficiently during all of the implementation and tailoring.

7.6.2 Perceived drawbacks

The thematic map of codes and themes of perceived drawbacks can be seen in Table 17, presented in the same manner as for perceived benefits previously. The order of themes displayed in the table is based on the number of coded answers from the open-ended questions relating to each theme.

This means that the theme displayed at the top of the table has the highest amount of coded answers from open-ended questions, and the following themes are displayed in falling order of number of coded answers. Where two themes share the same amount of coded answers from the open-ended questions, the number of codes from observations and interviews decides the order.

Table 17. Codes and themes of perceived drawbacks at Auto.

Codes from open-ended questions	**Codes from observations and interviews**	**Themes**
Too many meetings and too much administration [22]	Perceived inefficiency [4]	*Too much time spent in meetings*
Too slow [3]		
Too rigid, not Agile [13]	Difficult to apply to hardware development [3]	*Not tailored to the work process*
Not enough tailoring to our way of working [6]		
Too much according to SAFe [2]		
Only our department works like this [9]	Difficult to coordinate between trains and departments [7]	*Clashes with surrounding organizations*
Coordination between trains and projects [3]		
Coordination between different sites [3]		
Planning and estimation difficulties [7]	Not enough long term focus, short term focus is rewarded [5]	*Long-term planning difficulties*
Losing the whole picture [3]	Struggle between sense of control and seeking bigger picture [4]	
Unclear responsibilities [4]	Expectations on stakeholders, PMs, and managers [6]	*Unclear responsibilities*
	Who takes the initiative? [3]	
Stressful and frustrating [4]	Estimations cause stress [3]	*Stress*
Not enough commitment [3]	Less overall sense of responsibility [3]	*Lacking sense of overall responsibility*
Limited interaction between teams [1]	Not enough ad-hoc communication [5]	*Less ad-hoc coordination*
	Not fun if you are shy [3]	*Shyness*

The themes created for perceived drawbacks are explained below. Numbers within brackets show the number of answers from open-ended questions, as well as observations and interviews that belong to the same code.

Too much time spent in meetings

The drawback mentioned most in the open-ended survey answers was frustration regarding the amount of time spent in meetings. Specifically,

many complained about having to attend too many meetings [22]. One example of such an answer was: "I have to spend huge amounts of time in meetings that do not concern me". Some perceived that spending time in meetings made them too slow [3] and this frustrated them: "Longer lead times […] create a lot of overhead and frustration".

In interviews, respondents expressed how some coworkers complained about the amount of time spent in planning meetings. They perceived an inefficiency [4] due to the amount of time spent in meetings. As an example, Eve Dev expressed her doubts in the following way: "I don't know whether all this planning time we spend is so efficient. [We] should be able to cut down on the time spent on those planning activities".

Not tailored to the work process

The second most common perceived drawback from the open-ended questions was that the routines seemed to be "not Agile" [13]. Apart from that, several respondents complained that there was "not enough tailoring to our process" [7]. Since the organization had worked Agile for several years, several employees perceived that added routines were, to some extent, a step backward. Especially, the long-term planning was not tailored to the process they were accustomed to with short-term plans, one sprint at a time. Two answers suggested that the new routines were "too much in accordance with SAFe" [2].

In interviews, developers said that they perceived this way of working difficult to apply to hardware development [3]. One of the software developers expressed his thoughts in the following way:

> Those who are hardware developers have a harder time taking it all in. [I] often hear that they struggle with breaking down activities and delivering parts on an ongoing basis. They haven't worked like that before. (Andrew Dev)

Clashes with surrounding organizations

One area of frustration found in answers to the open-ended questions was: "Only our department works like this" [9]. This meant that other departments had other routines when it came to planning and meetings that differed from Auto. Another department could, for example, have shorter planning horizons. That could pose a problem when Auto had dependencies on teams in that department, as they might not be able to commit to deadlines.

SAFe prescribes much of how teams within an Agile Release Train should coordinate. Another problem highlighted in the answers to the open-ended questions was that there were no routines suggested for "co-

ordination between trains and projects" [3] or for "coordination between different sites" [3].

This was also expressed in interviews where respondents complained about the difficulties in coordinating between the three trains at Auto and teams in other departments [7]. One example of this has already been presented above, where Tod RTE explained this problem in relation to PI 6 at Auto:

> Where we have problems is when we have dependencies on the other trains. Within the train, it works relatively well ... It becomes a little more difficult when you talk to another train, and they like 'Aaah, but we have our plans here, you see' [laughter]. I would say that it goes quite well when we find dependencies within the train, but between the trains, we can be more efficient. (Tod RTE)

Long-term planning difficulties

Despite many perceived improvements in planning and increased planning precision, there were several answers to the open-ended questions where "planning and estimation difficulties" [7] were expressed. Some examples from the written answers are: "Difficult to plan for 9 weeks" and "Tasks are difficult to estimate". Another problem mentioned was the risk of losing sight of the whole picture [3]: "Difficult for the teams to understand the strategy as a whole and why we need to do something".

A perceived drawback mentioned in the interviews by two respondents was the problem of not having enough long term focus since short term focus is rewarded [5]. Oswald RTE expressed the problem as: "It is important not to make the mistake of only solving what is closest to you, they are urgent problems, and forget about the long term". He also added:

> Maybe this is not only an Agile problem, but I think it is amplified a little by the Agile ways of working because you… yes, you get… you are kind of rewarded for the number of tasks you solve. (Oswald RTE)

In one of the interviews and in discussions with employees during PI planning, respondents speculated that there would probably always be a struggle between feeling in control of the situation and, the need to understand the bigger picture [4].

The respondents felt that it could be difficult to plan for a longer period because you might have to spend a lot of energy on making longer and detailed plans. Plans that might need to be discarded if something unforeseen happened. If the plans covered a shorter period, there was a

small risk of having to discard them due to changes, and you could get a sense of control of what will happen in the future.

Unclear responsibilities

Some informants from the open-ended questions perceived that there were "unclear responsibilities" [4]. They perceived the problem as a clash between new and old roles.

In interviews, one question that arose from respondents when it came to added routines was: "Who takes the initiative?" [3]. Previously, when a Project Manager oversaw several teams, the obvious answer was that the Project Manager should take the initiative. Now, when many teams were organized into Agile Release Trains, with several projects being worked on in every train, it was not as easy to identify who it should be. This problem related to another area discussed in interviews regarding what expectations there were on stakeholders, Project Managers, and Managers [6]. For team members, Scrum Masters, and Product Owners, expectations are described in detail in SAFe (Scaled Agile, 2020). That is not the case with other roles in the organization. Those with roles not described in SAFe perceived their responsibilities as unclear.

Stress

Although several informants and respondents mentioned the benefit of perceived *stress relief*, some answers to the open-ended questions acknowledged a climate that was "stressful and frustrating" [4]. Respondents discussed that there was a risk of having too much short-term focus. With demonstrations of results after each sprint, short-term results were often rewarded.

Although Auto expressed that the goal of having a less stressful workplace had been reached, as Esmond Mgr pointed out in the quote above, answers from respondents showed that some still experienced stress. Especially, as put forth in interviews, when it came to the constant work of presenting estimations [3]. Andrew Dev explained: "We estimate and estimate, and it can feel stressful to constantly have to back new estimates up. Sometimes the estimations are very detailed".

Lacking overall sense of responsibility

Some answers to the open-ended questions showed that employees perceived that there was "not enough commitment" [2].

In interviews and during discussions at on-site observations, respondents talked about perceived sense of less overall responsibility [3] for the products. This was especially negative for Auto, since one of the stated

goals for implementing new routines was to increase the sense of responsibility for the products. Instead of having a too narrow focus on a specific problem, the intention was that better coordination and thereby better overview of problems and consequences would lead to an improved sense of product responsibility. However, according to Tod RTE, that did not happen.

> So this was something we thought would help with that, but it hasn't actually. Rather the opposite. We have more dropped balls that no one picks up. We have some problems with things being forgotten. If it does not fit perfectly with a Product Owner's backlog, we have a problem with people not picking them up. (Tod RTE)

If anything, Tod RTE thought that the problem of less overall sense of responsibility had even increased.

Less ad-hoc coordination

One answer to the open-ended questions was the perceived "limited interaction between teams" [1]. In interviews, respondents admitted that there was a problem with too little ad-hoc communication [5].

Some managers claimed that they had tried to deal with this for some time – that teams did not take the time to talk to other teams. However, Martin Dev confirmed that the problem remained: "We have become tighter as a team, but at the same time I think we communicate less with other teams". The problem, as they saw it, was that they discovered problems during the Scrum of Scrums, that could have been identified earlier. This might have to do with people who perceived that they attended too many meetings already. There might be resistance to spending even more time and effort on ad-hoc meetings.

Shyness

Although not mentioned in any of the answers to the open-ended questions, interview answers revealed that conducting some of the steps of the inter-team coordination routines was uncomfortable for those who were shy [3]. In one of the interviews, a respondent explained:

> It's really not fun, especially if you are a little shy like I am. If you are shy, SAFe [routines for inter-team coordination] is not any fun for you. You have been to our PI planning; you know at the end when we all present our plan for the train and other people? I know several that would never ever do that. To stand in front of everyone. (Nicole Dev)

She was referring to steps that some of the inter-team coordination routines involve many people. In Scrum of Scrums, all Scrum Masters and

the Release Train Engineer participated, but often even more people attended. During the final part of the PI planning event, plans were presented to all teams, managers, and stakeholders, which meant a large group of people. The negative effects of shyness, according to Nicole Dev, was that fewer people spoke up. Before that, team members only needed to speak up within their own team with people they had worked with for some time and trusted. However, with the new coordination routines there were many more participants, some of which team members did not know, and that might make them afraid to speak up.

7.7 Changes to team autonomy

This section presents changes to team autonomy as they were perceived by employees at Auto. The data collected for this part consists of interview answers to questions specifically regarding perceived changes to team autonomy. As described in the *Research method* chapter (5.7), the Agile principles from the Agile Manifesto (Beck et al., 2001) are used for analysis. The perceived changes to team autonomy are analyzed based on whether they support or conflict with the Agile principles.

7.7.1 Choosing features

Tod RTE expressed that what he had heard from teams in his Agile Release Train was that autonomy had become more limited in one area. Teams could not choose features as easily as before.

> I have understood from the teams that they think they are getting too detailed requirements … they are not involved in breaking them down… [therefore] they have a hard time really taking responsibility for [the feature]. They have a hard time to really feel for the requirements when they are already so detailed when they get them. (Tod RTE)

This view was confirmed by Baptiste Dev, who complained that this loss of autonomy also made work less joyful.

> Before, it was easier to choose whatever feature you wanted to work with … We don't get to choose much now, but it's way more dedicated to a specific area, and that's both good and bad, I would say. I understand more of the actual contents of what we do when we're in more narrow and more insight into the specific things that we're doing, but on the other hand, it's not as fun when you're too focused on one narrow area. (Baptiste Dev)

Delimiting the possibilities of choosing what to work with is not in accordance with AP 5 (*Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done*) or AP 11 (*The*

best architectures, requirements, and designs emerge from self-organizing teams). This impact on teams is thereby in contrast with these two Agile principles that describe the support and freedom intended for self-organizing teams.

7.7.2 Signaling limitations and having long-term goals

Marcus SM claimed that autonomy improved, since it was now easier to present limitations on how much work a team could accept for a PI. Marcus SM also reflected that a lack of overview and transparency may have limited teams when it comes to knowing how much they can decide for themselves.

> I think that we are not as limited as we previously thought. One of the things that have made us… that made this planning a little easier was, among other things, that we could signal what limitations we had. We've got more autonomy, I would say. (Marcus SM)

John SMPO, who was both the Product Owner and Scrum Master in his team, explained how much the teams differed when it came to long-term goals. He claimed that they had much autonomy, since his team knew what was expected of them for a longer period of time.

> We know where we are going … we have a clear picture of 'we will deliver this in a year'. There is no one else who writes work packages or features for us. So we can decide much more ourselves. Other teams have a lot more projects that push in stuff, so you can't say that they are self-governing. (John SMPO)

Signaling limitations and establishing clear long-term goals for the team is in accordance with AP 8 (*Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely*). At Auto, these changes to team autonomy support work in accordance with Agile principles.

7.7.3 Receiving and giving help

Baptiste Dev also claimed that autonomy had increased thanks to the improved overview and transparency towards other teams.

> We have more autonomy when it comes to receiving help or helping others as well. (Baptiste Dev)

Receiving and giving help is in accordance with AP 9 (*Continuous attention to technical excellence and good design enhances agility*), since much of the help is targeted towards technical design and development.

7.7.4 Diverse views

Regarding the change in team autonomy due to the implementation of inter-team coordination routines, answers from respondents were quite disparate. Some perceived that team autonomy had increased, some experienced that it had decreased. One of the identified benefits at Auto was *stress relief*. This might be due to an increase in team autonomy, at least according to a study by Meier et al. (2018) who concluded that self-organizing was the single most important factor for lower levels of stress. If this is in correlation, it is no surprise that one of the drawbacks identified by some employees was the counterpart: *stress*.

One of the Release Train Engineers and one Scrum Master expressed the following:

> I would say that team autonomy is the same today as it was two years ago, before the implementation. (Oswald RTE)

> The team has about the same mandate for decision-making now as they had before, no major difference there. (Lisa SM)

With these two respondents, who claim there had been no change to team autonomy at all, it is evident that there were very diverse views of how autonomy had changed at Auto.

8 Case Bank

The different sections in this chapter follow the same order as the previous chapter. The goal of this chapter is to highlight findings that are particular to the Bank case.

Although Bank is a department within an international bank with offices all over Scandinavia, the Baltic countries, Great Britain and Germany, all employees working at the observed department are native Swedish speakers. Hence, all presentation slides, discussions, and interviews were performed in Swedish, which means that quotes and texts presented in this section have been translated into English. In the same manner, as with Case Auto, all respondents are anonymized and have been given other names. Instead of family names, abbreviations of their role are used. Four new abbreviations are used in this case compared to Case Auto. The first one is *Arch*, which is short for the role Systems Architect. The second is *PMgr*, which is short for the role of a Product Manager. The third is *Coach*, which is short for Agile Coach and, finally, *Test*, which is short for Software Tester. Table 18 presents the interview respondents and employees quoted at Bank.

Table 18. Quoted employees at Bank.

Alias in quotes	**Role at Bank**	**Interview date**
Andrew Coach	Agile coach	2017-04-25
Philip Arch	System Architect	2017-08-16
Annabelle RTE	Release Train Engineer	2017-08-16
James Dev	Software Developer	2017-12-18
Catherine Dev	Software Developer	2017-12-19
Peter SM	Scrum Master	2017-12-19
Elmer Test	Software Tester	2018-03-07
Jocelyn Dev	Software Developer	2018-03-07
Lance Mgr	Department Manager	Quotes from observations
Steven PMgr	Product Manager	Quotes from observations

First, some initial background information. Bank (pseudonym) is a software development department in one of the largest business banks in Sweden. The average number of years that employees have worked at Bank is 9.6 (SD = 7.5). The studied department introduced Agile ways of working in self-organizing teams during 2014 and started implementing inter-team coordination routines inspired by SAFe in early 2017. Before implementing these routines, the teams in the department were not much depending on each other. The need for further coordination arose from a

decision to develop a new version of an old system. This was a major undertaking where several teams needed to cooperate to develop the new system.

Bank is organized into one Agile Release Train. One person has the role of Release Train Engineer and coordinates the whole train. While setting up the Agile Release Train, teams were reorganized based on value streams instead of one team being responsible for one IT system:

> [The teams were] organized on the basis of systems rather than function or some kind of total value creation. ... it didn't really work, because we were constantly dependent on each other... it doesn't work in this world that one system is completely independent and can create a whole value. It's a chain of systems, so to solve it; we started looking at what value chains, or streams, we actually deliver. And that's really how we built our teams. (Peter SM)

The view on inter-team coordination routines suggested by SAFe and the reason for implementing the routines at Bank were explained by the Agile coach:

> After all, SAFe is a concoction of many other methods, really just a packaging as I see it. It doesn't add much that is new, but it is a good packaging as well as a shift for the organization to focus from one team to several teams. The methods are tried and tested, and we know they work, provided we give the right conditions […] SAFe is exactly what they say it is, but not everyone hears. It's a 'take what you think is relevant to you and use it if it brings benefits to your organization'. And that's exactly what we try to do. (Andrew Coach)

Figure 14 shows a timeline of the performed observations at Bank.

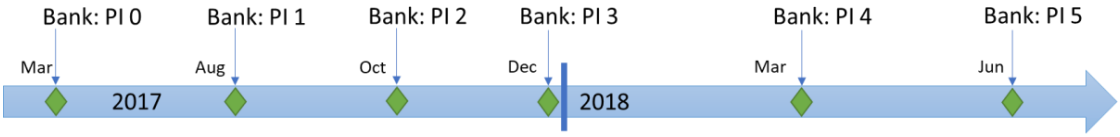

Figure 14. Timeline of PI planning events when data was collected at Bank

Below, the first act presents the planning and execution of PI 0, the second act PI 3, and the third act PI 5.

8.1 The First Act: PI planning for PI 0

Before the first act, the teams had worked separately for four sprints, each sprint between two and two-and-a-half weeks long. The implementation of inter-team coordination routines and joint planning was started during the transition from the fourth to the fifth sprint. Instead of calling this PI

planning, they named it *Big-room planning*, since managers at Bank did not want to label this as a pure SAFe implementation.

> Bank is a bit like reinventing the wheel just because you want to put your own labels on it. I'm not a friend of it. (Andrew Coach)

For the first PI planning organized at Bank, the department manager only allowed four hours for a PI planning workshop of which the Agile coach responded (somewhat irritated) that with so little time, they would only be able to plan for two sprints (sprint 5 and 6). Since the workshop was planned to be so short, it was renamed to *Mini Big room planning*. After the anger subsided, the Agile coach changed his view on the PI planning workshop and saw benefits from this approach: instead of learning to plan according to a strict cadence. He concluded that the teams would learn faster how to conduct PI planning workshops, since they would have the next workshop as early as five weeks after the first attempt. In other words, more PI planning events within a shorter timespan would lead to faster learning and better routines for planning, according to Andrew Coach.

On Tuesday, March 28, 2017, the Agile Release Train containing five teams (at the time) and different stakeholders met for their first PI planning event. During the first minutes of the planning workshop, a slide intended to explain the reason for performing PI planning was presented to all participants. The slide contained the following text (translated from Swedish):

"Why are we here? We are here to:

- reach consensus on what business and architectural goals are important for the period
- create a joint plan for the period, based on actual capacity."

Next, important dates were presented. The start date of the PI (this day) and the end date of the PI (five weeks ahead, the 9th of May) were shown together with the two sprint planning days during the PI. Also, two planned software releases during the period were presented.

Three different presenters gave their short view on the business context, product/solution vision, and architectural vision. This is all according to how SAFe prescribes the first part of the PI planning workshop, although it was performed in only two hours instead of the four hours that SAFe suggests (Scaled Agile, 2020).

This was immediately followed by two hours of team breakout where teams chose features to build from a prioritized product backlog present-

ed as post-it notes on a wall. Once during the team breakout, all Scrum Masters met for a short *SoS check-in*. In this meeting, all Scrum Masters answered 13 questions following a checklist to verify that all teams were on track with their planning. One example of a checklist questions was: "Have you identified your dependencies?" The checklist was put together by Andrew Coach by choosing a number of questions relevant for Bank from a much longer list prescribed by SAFe.

Each team planned and estimated their features for the PI period and presented dependencies between teams on a program board, as presented in Figure 15.

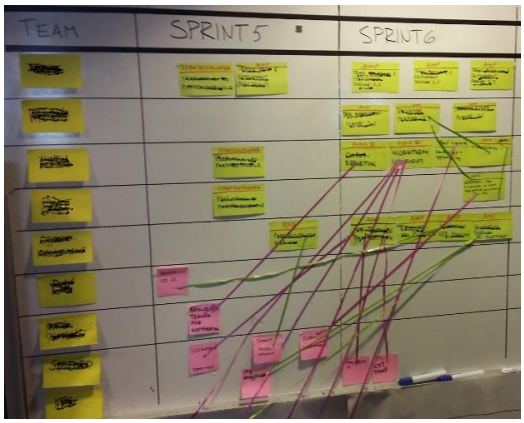

Figure 15. Program board after PI 0 at Bank.

The top five rows in Figure 2 show planned features for the five teams (the yellow post-it notes in columns "Sprint 5" and "Sprint 6"). The rows below the first five rows show teams, departments, and other organizations with dependencies to Bank. Only three dependencies (the cords between yellow post-it notes) are between teams in Bank. Fourteen dependencies (the cords between yellow and pink post-it notes) to Bank are not within the train, but outside the train.

At the end of the final fourth hour, teams presented their final plan. During five-minute presentations, each team explained in words, the chosen features and dependencies as depicted in Figure 15.

After a short wrap-up of the PI planning event, it was time to start developing features. The PI 0 period started the following morning with a detailed sprint planning event conducted separately for each team for sprint 5.

8.1.1 A static program board during the PI

According to SAFe, teams are supposed to update the board continuously during the PI planning workshop as soon as a new feature is planned for. At the end of the planning days, the program board should, therefore, be complete and show all known dependencies for the upcoming PI. Then, as work on the new PI begins, SAFe states that the "program board is often used during the Scrum of Scrums meetings to track dependencies, or it may not be maintained (manually) after that time. This depends upon the Agile project management tooling in place and the needs of the Agile Release Train" (Scaled Agile, 2020). This description is a clarification from the previous version of SAFe, 4.0 (Scaled Agile, 2016), which only expressed that "the program board may or may not be maintained after that time," meaning after the PI planning workshop.

When interviewing the Release Train Engineer, I asked whether Bank updated the program board during the PI period and received the following reply:

> We have not used the program board like that ... it has not worked out so good, I would say. We did use it when we were planning, but then we haven't really looked at it, I'm sorry to say. (Annabelle RTE)

The reply shows the intention to keep the program board updated, but failure in doing so was viewed as a failure in the implementation of the coordination routine.

8.1.2 Scrum of Scrums becomes a daily routine

Bank chose to conduct Scrum of Scrums in the same format as the daily standup meeting (Sutherland & Schwaber, 2013), with a fixed timebox of 15 minutes per day, always between 12.45 and 13.00. The time was chosen based on that many other kinds of meetings at Bank started at 13.00.

During the first part of PI 0, Scrum of Scrums was only conducted every second day, i.e., two to three times a week. This later evolved into a daily routine with more diverse participation:

> From the beginning it was just Scrum Masters and Product Owners, but it became clear that this was in everyone's interest, to hear a little about how things were going and risks emerging […], so this meeting invitation has grown wider and wider, so there is a big commitment and interest around [Scrum of Scrums] now. We are expanding it to have it every day now. (Annabelle RTE)

However, Bank experienced some difficulties regarding the focus of the meeting:

> Some just want to tell you what they do, to get a pat on the shoulder like "we're really good" which everyone deserves, but it sometimes shifts to the wrong focus. (Annabelle RTE)

The focus in the Scrum of Scrums was primarily to help between teams, and every Scrum Master mainly talked about what their team needed help with. Based on the obstacles in the single team, additional team members sometimes joined the Scrum of Scrums to explain the details of the problems further. Although the time frame was the same, meetings could be held in a variety of ways. For example, if some issues could not be solved during the Scrum of Scrums, parties involved stayed to resolve issues in a "meet after" whenever necessary.

8.1.3 Only one hour allowed for PI retrospective

The PI ended with a short retrospective meeting that in SAFe is called *Inspect and Adapt*. This spanned the whole PI period (sprints 5 and 6) and all team members joined in for a one-hour meeting. To use their own labels, Bank just called the meeting *Retro*. SAFe prescribes four hours (half-day) for this meeting, and Andrew Coach was very frustrated that Bank only allowed one hour. A presentation slide was jointly put together to summarize which areas needed improvements for the upcoming PI planning event. The slide contained the following (translated from Swedish):

"Preparational work with planned features:

- More detailed
- Anchored in the teams
- Better descriptions to understand the scope
- Clearer Milestones

The joint PI planning event:

- Too little time for planning
- Invite more people from the business side."

When Annabelle RTE measured and reported actual results versus plans to management after PI 0 was finished, only 60 percent of the planned, estimated features were completed.

8.2 The Second Act: PI planning for PI 3

During the first PIs (PI 0, PI 1, and PI 2), only two sprints were planned each time. During PI planning for PI 3, three sprints were planned. At the PI 4 planning workshop, Bank finally decided on a fixed cadence of four sprints per PI for future PIs.

> When we wanted to do the first PI planning, the reason why we only planned for two sprints was actually that we were not allowed to have a longer planning workshop, only four hours. Then later, I thought "well, that's actually great, [it] means [that] we will learn how to PI plan faster." That's why we have expanded the PI bit by bit, instead of having a long fixed cadence. (Andrew Coach)

When the PI planning event for PI 3 started on December 19, 2017, they were allowed one-and-a-half-day for planning. One team was split into two, which meant that Bank now consisted of six teams. An additional team employed at another department had now been moved to join the train in the same room, since the teams had identified several dependencies with them during the previous PI. Therefore, the Agile Release Train at Bank now consisted of seven teams.

Annabelle RTE, who was responsible for planning the PI planning event, expressed her worries only minutes before the start:

> Once again, the breakdown and detailing of features were made too late. The team got to see these too late, and some people hardly know anything about them. Our Product Owners are more like proxies; they do not own and control their backlog. The main problem is that there are rarely any pronounced benefits for the requested features. Product Owners aren't passionate about their features and can't explain why they are needed (Annabelle RTE)

Annabelle RTE opened the PI planning event, expressed the same goals of planning as in PI 0, and let the department manager continue to explain the business context. A Project Manager and the Systems Architect, Philip Arch, presented the upcoming intended work with the systems and architectural vision for PI 3. Philip Arch finished the presentation by adding: "Are you interested in R or Python? Let me know so I can arrange a course for you!"

Before starting team breakouts, Annabelle RTE and Andrew Coach reminded everyone of the importance of updating the program board continuously during planning. A small tailoring decision was also presented: except for team dependencies, the teams were supposed to mark which features belonged to a certain milestone delivery. In other words, it should be easy to understand which features would be part of a certain

delivery only by looking at the program board. Annabelle RTE finished this part by adding:

> I think you are all starting to see the value in using the program board now. If you have any suggestions on how it can be even more effective, please let me know! (Annabelle RTE)

During team breakouts, the Scrum Masters met in SoS check-in meetings three times on the first day and twice on the second day. In the final SoS check-in of the first day, angry voices were raised. Several teams had misunderstood when some important features needed to be finished, and a lot of planning time had been wasted. Everyone agreed that this problem would not have occurred if the features had been put up on the program board first. Annabelle expressed that the program board should always be the "master" planning tool. She realized that this should have been better communicated at the beginning of the PI planning event.

At the final wrap-up of the PI planning event, both Annabelle RTE and the department manager Lance Mgr summed up with some final words:

> Everyone is starting to get the hang of this [PI planning] now; it's like a standard routine. I feel superfluous. (Annabelle RTE)

> From a larger perspective, I can see how we get more and more value of this [way of working]. More transparency, more value in what we believe in, precision. At a higher level, we also see the obstacles. I feel confident in working with you all. (Lance Mgr)

As before, Annabelle RTE measured and reported plans versus actual results after PI 3 was completed. From having only 60 percent of the planned features done after PI 0, results had been continuously improving. After PI 1, 72 percent of planned features were done, and after PI 2 the result was 77 percent. Now, after PI 3, 82 percent of the planned features were done.

8.2.1 Inventing the mid-sprint review routine

Bank chose to add another routine to follow up progress: the "mid-sprint review". It was conducted on the sixth or seventh day of each twelve-day sprint. In this meeting, every Scrum Master reported and discussed the progress of the teams' objectives, and the program board was updated accordingly.

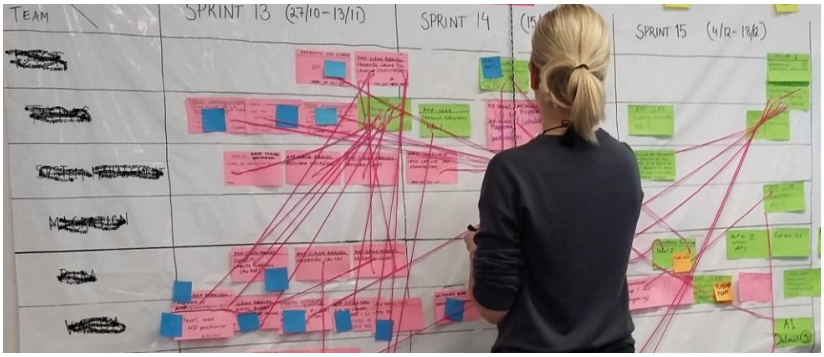

Figure 16. Updating the program board during a mid-sprint review meeting at Bank.

In this meeting, all solved dependencies were marked with a small blue post-it note (see Figure 16). Features with dependencies that were delayed, they were moved forward on the program board to show where they were now planned to be resolved. Also, the dependent features that were moved were tagged with an orange post-it note to show a change from what the board had looked like at the end of the PI planning event.

8.2.2 No Inspect and Adapt routine allowed

According to SAFe, the Agile Release Train is supposed to perform an *Inspect and Adapt* routine, renamed to *Retro* at Bank, in a joint meeting with all teams at the end of the PI. This meeting involves retrospective discussions on possible improvements based on experiences from the whole PI period. The expected benefits of doing this, according to SAFe (Scaled Agile, 2020), is increased knowledge about the whole business for everyone and a better understanding of why things are done.

Bank did not perform the *Inspect and Adapt* routine at the end of PI 3. When asked about the reason for this, Annabelle RTE simply replied that they did not have enough time to do it. The Department Manager, Lance Mgr prioritized more hours of work instead of conducting a *Retro*.

8.3 The Third Act: PI planning for PI 5

On June 4, 2018, Bank consisted of 7 teams with 42 employed developers and testers, which means that the average team consisted of 6 team members. During this last observation, PI 5, four sprints had been used as PI length for two PI periods, and Bank did not see any benefits in reaching for more extended planning periods, according to Annabelle RTE. As usual, Annabelle RTE opened the PI planning event and expressed the same planning goals as in all previous PI planning events. She then let a senior Division Manager continue to explain the business context from a

management perspective. The Product Manager and one of the Scrum Masters presented the product/solution and architectural vision for PI 5. Steven PMgr finished the presentation by saying:

> Previously we went through and talked about upcoming features in detail, but now we have involved the teams in advance. You have all helped us in detailing the requirements and, by the way, leave time in your plans for doing so during the PI for the next PI as well ... Since we have made an analysis of all features, which you can read about in the documents here, we will only describe them briefly now. Another difference is that some dependencies are explicitly stated in the feature description now, as we identified them during the preparations. Some will own several features and will then be responsible for coordinating with the right team. (Steven PMgr)

Before team breakouts, Annabelle RTE clearly pointed out that the number one planning and coordination artifact was the program board.

> Remember to start by highlighting features and their dependencies on the program board before you begin to plan the feature. It may even be better to seek out the dependent team to plan details with another team directly. Try to have a wider mindset: what will be the effects of what we do? Come to me if you think we should have more stakeholders at refinement meetings or demos (Annabelle RTE)

This can also be seen in the following short discussion:

> [To the Scrum Master in team 1 and 2:] Sit down for fifteen minutes right after this and find out which team will do what. (Steven PMgr)

> Now, go up to the program board instead and put up a note to show your dependencies there! (Annabelle RTE)

All teams finalized their plans during the final team breakout and presented them in the final joint meeting. One developer expressed that he perceived planning together made it more efficient:

> Although we are pressured and feel a lack of time, this way [all teams planning at the same time] makes it possible to get the planning done on time. I didn't think so when we started again after lunch. (James Dev)

Once again, Annabelle RTE measured and reported plans versus actual results after the PI 5 period was completed. The results were the same for both PI 4 and PI 5: 87 percent of the planned features were developed.

At the end of PI 5, the teams were allowed two hours for a *Retro*, their name for the *Inspect and Adapt* meeting.

8.3.1 One team responsible for one complete feature

A new way of coordinating and dividing work for PI 5 was to have *full feature responsibility*. Instead of only displaying which features having dependencies on other features, as done previously, this meant that one team now instead owned the responsibility for each set of conjoined features.

> I will say this again: instead of doing as we did previously when we had dependencies between maybe four teams, from now on, we try to have one team responsible for all the dependencies for a whole feature. We are testing this now and see if it works better. (Peter SM)

This still meant that the program board should depict dependencies between features, but if, for example, four teams would have features dependent on each other, one of these four teams would have full responsibility for following up the work with the other three teams to check on solved dependencies.

The above quote also describes the experimental approach emerging at Bank. A new routine was hereby invented based on discussions, retrospectives, and new ideas.

8.4 Identified Institutional logics

The decisions made when tailoring the inter-team coordination routines and artifacts show the internal dynamics at Bank. In observing and reflecting on routines and decisions on tailoring from the case, interactions of these dynamics made it possible to identify signs of institutional orders. The way these tailoring decisions manifested in each institutional order pointed to the dominant logic. Table 19 summarizes the different dimensions at Bank that present the beliefs and values of specific institutional logics, creating an institutional logic ideal type consisting of Agile toolbox logic and a Hybrid (Resource/Flow) efficiency logic.

Table 19. Institutional logic ideal type at Bank

Dimension	Agile toolbox logic	Hybrid (Resource/Flow) efficiency logic
Root Metaphor	Agile routines as tools	Efficiency as utilization of man-hours *moving towards* optimization of workflow
Sources of Legitimacy	Unity of improvement needs	Personal efficiency *moving towards* Team efficiency
Sources of Authority	Commitment to team decisions	Management *and* Measurements
Sources of Identity	Attentiveness and Responsiveness	Competence and flexibility (Flow efficiency logic)
Basis of Attention	Routine inefficiencies	Resource availability *moving towards* Workflow delays
Basis of Strategy	Invent and adapt routines	Utilize resources *moving towards* optimizing flow, reduce time waste

When asked about implementing inter-team coordination routines and the view on SAFe, Andrew Coach responded in the following way:

> …SAFe is exactly what they say it is, but not everyone hears. It's a "take what you think is relevant to you and use it if it brings benefits to your organization". And that's exactly what we try to do. (Andrew Coach)

This quote is strong evidence of a dominating *Agile toolbox logic* at Bank. Andrew Coach further expressed how decisions were made when implementing inter-team coordination routines at Bank, such as *Big room planning*. He expressed that Bank often 'reinvented the wheel' by putting their own label on things, such as renaming PI planning. That is also an example of an Agile toolbox logic where, instead of pointing to a successful method being adopted, routines are renamed and adapted to fit the organization. Bank also invented new routines such as the *mid-sprint review* and *full feature responsibility*, which are also examples of an Agile toolbox logic. Tools are not only chosen from a toolbox but also invented.

The following quote regarding how Agile ways of working where introduced historically was expressed in an interview with Peter SM.

> We were probably first out in the organization [to implement Agile ways of working], I think, and we had to work a lot under the radar. We obviously reported what the project method required of us with lots of decision points and a lot of reports… [but] without their knowledge, we actually worked very Agile. (Peter SM)

By implementing and trying out different routines for Bank even without management knowledge, this is a typical example of *Agile toolbox logic* even in the past.

Analyzing whether a dominating resource efficiency or flow efficiency logic was enacted turned out to be more problematic. A clear pattern could be recognized in many of the initial tailoring decisions at Bank: allowing shorter time for meetings. The initial decision to allow only four hours for PI planning was not based on a certain reason; only that four hours was the allotted time for non-productive work. The fear of wasting hours on non-productive work was the recurring basis for several decisions.

This is a clear sign of resource efficiency logic, making sure all resources are planned for and utilized each minute. Too much time for planning was considered unproductive at Bank. Another example was the *Inspect and Adapt* meeting at the end of a PI period. Time for reflection in an Inspect and Adapt meeting was abandoned due to lack of time in PI 3. It was not until PI 5 that Bank was allowed more than an hour for reflections and process improvements. This is evidence of Bank moving towards flow efficiency logic.

The logic behind resource efficiency is often the idea of productivity: if every resource is planned in detail and utilized, the work must become productive. Andrew Coach reflected that the most important thing for Bank was to predict who was going to do what, since there was a prevailing fear of *not knowing* whether resources were utilized.

> If I were to say one thing that is most important for Bank, it is predictability. Everyone screams about productivity all the time, but I don't think that's the truth. I think predictability is more important. (Andrew Coach)

Another example of resource efficiency logic was a strong focus on the reporting hierarchy.

> I don't really understand the point with a lot of the reporting. It's like, "can we report that this makes us compliant? Fine! Let's report this", even if we don't know if anyone reads it … We have to keep track of our capacity according to management. But we don't know it. So that's why an order has been issued that all teams must measure their lead times and also their velocity ... and everything is reported and aggregated. Sigh! Reports and reports and reports. You hear how it sounds. (Andrew Coach)

Regarding *sources of authority*, this seems to point to measurements being in focus. However, as Andrew Coach explained, since they could not use the provided information, they just did as they were told, i.e., *management* was the source of authority. Therefore, this reporting pattern followed resource efficiency logic. And since detailed plans were frequently reported, no one wanted to report that resources were not being utilized.

Time not planned for productive work would look like unproductive time. Hence the initial tailoring decisions of shorter time for PI planning and the abandoned Inspect and Adapt meeting.

At the same time, some evidence indicated that flow efficiency logic was the prevailing logic at Bank. One example was the statement by Annabelle RTE that "the teams were reorganized based on value streams instead of one team responsible for one IT system". This decision was based on the prescribed way of working in SAFe to turn the organization to flow efficiency logic. Team efficiency became more important than personal efficiency. It was also a move towards optimization of workflow, rather than utilization of man-hours.

Another example of the shift towards flow efficiency logic was how Scrum of Scrums were tailored. After having a meeting twice or three times a week, the meeting was held daily, often with more roles attending than just Scrum Masters and Product Owners. This showed that the attention at Bank moved towards solving workflow delays.

Philip Arch expressed the following during PI planning of PI 3: "Are you interested in R or Python? Let me know so I can arrange a course for you!". This initiative of broadening expertise among employees is an example of *competence and flexibility* as a source of identity (flow efficiency logic), rather than clearly defined roles.

To conclude: the investigated period shows a shift towards flow optimization at Bank. During the observed period, Bank was influenced by two competing logics at the same time. A hybrid institutional logic, if you will, consisting of two competing logics. This is not unheard of in previous research. Doležel (2018) gives an example from Central Europe on a larger scale. A common institutional logic in the West is capitalism. In other parts of the world, the prevailing institutional logic is Communism. During the first few years after the fall of the Iron Curtain, the ex-Communist countries in Central Europe experienced and enacted both of these conflicting logics.

During the observed period, resource efficiency logic was first dominant but this evolved more and more towards flow efficiency logic. As can be seen in this example, even in clear-cut cases there is rarely complete adherence to a specific logic (Ferrer Conill, 2018).

8.5 Analysis of coordination routines

Below, both the *how* and *why* aspects of the performed and tailored inter-team coordination routines are analyzed. Table 20 summarizes differences between the prescribed performance of inter-team coordination routines according to SAFe and how routines were performed at Bank.

Table 20. Routines at Bank from a tailoring perspective

Routine	**According to SAFe**	**Bank**
PI planning	Set cadence, PI plan every 8-12 weeks.	Started with 5 weeks and finally set at 9 weeks' cadence.
	4*2-week sprints and IP sprint (2 weeks).	First: 2*12-day sprints. Finally: 4*12-day sprints and 2-day IP sprint.
	Duration: 2 days.	Duration: From 0.5 to 1.5 days.
Scrum of Scrums	Twice weekly.	Daily + mid-sprint review.
	Duration: 30 minutes and "meet after".	Duration: 15 min, sometimes "meet after".
	Attendees: RTE, Scrum Masters, and possibly stake-holders.	Attendees: RTE, Scrum Masters, Product Owners, some stakeholders, and some team members.
Program board	Contains: Feature delivery dates, feature dependencies, and milestones.	Contains: Feature delivery dates, feature dependencies, and milestones.
	May, or may not, be maintained during PI.	Updated every sprint planning day and during mid-sprint review.

Looking first at the PI planning routine, teams at Bank were first only allowed four hours, but this was later increased each PI to allow for planning of more sprints ahead, from two to finally four sprints. Table 21 shows the time and intervals of the PI planning and the PI Period.

Table 21. PI planning data from Bank at PI 0, PI 3, and PI 5.

	Bank PI 0	**Bank PI 3**	**Bank PI 5**
Number of weeks per PI	5	8	11
Number of sprints per PI	2	3	4
Sprint length (weeks)	2.5	2.5	2.5
Length of final Inspect and Adapt meeting	1 hour	None	2 hours
PI planning workshop (hours)	4	12	12
Team breakout time (hours)	2.1	7.5	7.5
Team breakout time (percent):	52.5%	62.5%	62.5%

To summarize the tailoring conducted at Bank relative to prescriptions from SAFe (Scaled Agile, 2020):

PI Planning

- Cadence – increasing number of sprints in the first PIs to learn faster
- An increasing amount of team breakout time

Scrum of Scrums

- Every day
- Different stakeholders attending each meeting

Use and management of the program board

- Additional notes to show 1) what is done and 2) delayed work

The PI planning routine was tailored from the very start and kept on being tailored along the way based on feedback from participants. At Bank, the intended reason for conducting a PI planning event was described as: to create consensus on what business and architectural goals are important for the period and to create a joint plan for the period based on actual capacity. This coincides roughly with how SAFe (Scaled Agile, 2020) described the reason: aligning all the teams on the Agile Release Train to a shared mission and vision. However, another reason stated in SAFe is to train teams to plan according to a set cadence to learn about actual capacity. According to SAFe, this will lead to better estimates and better predictions in planning. Bank instead started with a short PI period and did not set on a fixed cadence until PI 4. Instead, the reason for prolonging PI periods gradually was to get more PI planning events done in a short time and thereby to increase the number of planning events. Instead of training teams on planning for a set period, Bank trained the teams to become better at performing the PI planning event itself. Annabelle RTE expressed feeling superfluous already at the planning event of PI 3. An additional purpose of PI planning can be seen in the quote from Philip Arch in PI 3. He suggested setting up courses for the developers at Bank. Using PI planning to impose further knowledge sharing of technical competence, not only regarding mission and vision, is not suggested in SAFe.

Another routine to analyze is Scrum of Scrums. Bank chose to conduct Scrum of Scrums in the same format as the daily standup meeting (Sutherland and Schwaber, 2013), with a fixed timebox of 15 minutes per day, always between 12.45 and 13.00. The time was chosen based on that

many other kinds of meetings at Bank started at 13.00. The focus in the Scrum of Scrums was primarily to help out between teams, and every Scrum Master mainly talked about what their team needed help with. Also, Bank chose to add another routine to follow up progress: the *mid-sprint review* always conducted on the sixth or seventh day of the twelve-day sprint. Based on the obstacles in the single team, additional team members sometimes joined the Scrum of Scrums to explain the details of the problems further. Although the time frame was the same for each Scrum of Scrums, the meetings were held in different ways. Different stakeholders attended each meeting and, if some issues could not be solved during the Scrum of Scrums, parties involved stayed to resolve issues in a "meet after" whenever necessary. All parties at Bank saw the Scrum of Scrums as a way to solve issues between teams based on dependencies just as prescribed in SAFe.

The next routine to investigate is the use of the program board. Instead of updating the program board during Scrum of Scrums, Bank updated the board during the first day of every new sprint and during the *mid-sprint review*. In addition to what SAFe prescribes, color-coded notes were used to make the program board more value-adding. Implemented features were marked with blue notes and postponed features with orange notes. Regarding purpose, all roles had a similar view on why the program board, and its content, should be used. To Bank, however, a small shift in purpose was the initial view of the program board as "just another tool" which later became the central overview of the final plan, the *master*. Teams were encouraged to begin at the program board before they continued planning. This showed the increasing importance of the program board at Bank.

8.5.1 Dependency analysis

The proper starting point for analyzing coordination is to investigate what kind of dependencies the organization is facing, according to coordination theory (Malone & Crowston, 1994). In this section, dependencies are analyzed from three different perspectives.

One of the System Architects explained the background of the IT environment and the need for coordination:

> We have had separate administrations of systems where there has been a team that manages this system, a group that manages that system […] Now, we can see the team dependencies. You become very dependent on "you

> have to deliver this first, and then you can deliver your things after that". (Philip Arch)

Viewing this statement from a coordination theory perspective, it expresses a *producer-consumer dependency* issue. Especially, it puts forth the problems of two sub-dependencies. One is *transfer*, ensuring that the output is available to the consumer when needed. The other is *precedence,* making sure the actor performing the second task learns when the resource is available, and the task can be started.

A second way of analyzing dependencies is to categorize them in terms of workflow between actors based on Thompson's (1967) classification (described in chapter 3). It seems that most dependencies could be described as *pooled interdependencies* or *sequential dependencies*. As interview answers showed, different views were expressed on which kind was more prevalent. Peter SM considered sequential dependencies to be the dominant type of dependency issues:

> Where we are today […], it is mostly a problem of sequence, I would say. Actually, when I think about it, I would probably say that most of our dependencies are problems of sequence at the Bank. (Peter SM)

But according to Annabelle RTE, the most common problem with dependencies was rather the lack of competence, implying pooled dependencies to be the most common problem:

> No, I think the problem is that they don't know. So getting someone else to learn how to do something is a bigger job than letting someone else do it. (Annabelle RTE)

A third important area to analyze regarding dependencies is static versus emerging dependency issues. In other words: how much coordination can be solved in advance with good planning? From a qualitative perspective, three respondents expressed their views on this area. When asked about to what extent dependencies emerged during the PI period, versus identified during PI planning, the following answers were given:

> There are probably at least as many emerging as planned, I would say, but they are not the same size. The emerging ones are smaller. (Philip Arch)

> 30 percent, maybe? I would guess something like that, but unfortunately, we haven't counted them. (Annabelle RTE)

> It varies a lot between the teams. Some teams, I think, have maybe 20-30 percent [emerging dependencies], while others are around 10 percent. (Peter SM)

These three answers do not give a clear picture of the amount and importance of emerging versus planned dependency issues. They do, however, show that emerging dependencies are common and that coordination routines to handle them are of importance to Bank. This confirms the importance of coordination mechanisms to manage emerging dependency issues, as raised by Jarzabkowski et al. (2012) and Okhuysen and Bechky (2009). The importance of managing emerging dependencies might also be an explanation as to why Bank invented a new routine: the mid-sprint review. The mid-sprint review was specifically intended to solve the problem of emerging dependencies. According to Bank, the routines prescribed by SAFe (Scaled Agile, 2020) were not helpful enough, hence the invented routine.

8.6 Perceived impacts

At the beginning of the PI planning event for PI 5 at Bank, a paper questionnaire was handed out to all participants to be filled out. The full questionnaire can be seen in Appendix B. Several parts of the questionnaire were inspired by Laanti et al. (2011), and the first section was intended to understand the employees' views of the different routines for inter-team coordination.

Statements with Likert scales where used, ranging from 1 to 5, where 3.0 can be considered neutral. Three coordination routines were investigated in the first section, and the results are presented in Table 22. At Bank, 45 answers were collected from survey respondents out of 52 employees (a response rate of 86.5 percent).

Table 22. Perceived impacts of inter-team coordination routines at Bank

Survey Question (Likert scale 1-5: 1=Not true, 5=Very true)	**Mean**	**SD**	**Median**	**Mode**
The PI planning gives me a good overview of our work	3.9	0.5	4	4
The Program Board is very helpful for coordination between teams	3.3	0.9	3	4
The Scrum of Scrums (SoS) meetings solve inter-team coordination problems	3.4	0.8	4	4
There are many problems with working in a large-scale setting as ours, using SAFe	3.2	0.9	3	4
I would like to go back to the old way of working (instead of working according to SAFe)	1.9	1.0	2	1

As Table 22 shows, the view of how useful the PI planning event was averaged at 3.9, which shows a positive view of its usefulness. The view of Scrum of Scrums averaged at 3.3 and the use of the program board at 3.4, which shows a mildly positive view of the usefulness of these parts as well.

The average of the fourth statement (*There are many problems with working in a large-scale setting like ours, using SAFe*) was 3.2, which is slightly above neutral. The final statement (*I would like to go back to the old way of working (instead of working according to SAFe))* showed an average of 1.9 (and a mode of 1). This shows that most employees would not like to go back to the way it was before the implementation of the inter-team coordination routines.

8.6.1 Perceived benefits

As explained previously in the research methods chapter (5), the first step of the analysis was to investigate all open-ended answers from Bank to familiarize myself with the data (Braun & Clarke, 2006).

The second step was to generate initial codes by searching for patterns and grouping together answers. The subsequent steps resulted in a definition of themes. The themes can be seen in the right-hand column of the thematic map in Table 23.

Table 23. Codes and themes of perceived benefits at Bank.

Codes from open-ended questions	**Codes from observations and interviews**	**Themes**
Overview [14]	Overview [7]	*Overview and transparency*
Visibility/Transparency [4]	Transparent goal and vision [3]	
Common goal [2]	Collocation [2]	
Cooperation/ coordination/consensus [8]	Feature ownership across teams [3]	*Coordination and cooperation*
Communication [3]	Team stability [2]	
Speed [1]	Frequent PI planning events increase the speed [3]	*Speed, focus and efficiency*
Productivity [1]	Finish planning on time [3]	
Focus [1]		
Great fun! [1]	Knowledge sharing [3]	*Employee motivation*
	Involving teams before PI planning [2]	
	Increasing achieved versus planned work [6]	*Planning Precision*
	Better promises to management [2]	
	Many roles attending meetings [4]	*Less dependent on individuals*
	Less dependent on individuals [3]	

Numbers within brackets show the number of occurrences for each code, in the same manner as presented for case Auto. To give an example from the answers to the open-ended questions, the code "overview" comes from answers such as "better overview", "much better overview", "overview is better", and simply "overview".

Overview and transparency

The two most common answers to the open-ended questions were versions of "better overview" [12] and "improved visibility/transparency" [4]. Two answers relating to overview and transparency were about perceiving a clear "common goal" [2].

In interviews, five respondents talked about a better overview [7], such as Catherine Dev: "[It has] really been an improvement to get this whole overview and get managers to actually see what we can do and cannot [do]". Lance Mgr suggested that the improved overview had another benefit: "At a higher level, we now see the obstacles". Another benefit expressed in interviews and during observations was the increased

transparency of goals and vision [3]. Lance Mgr concluded at one PI planning event that Bank was now: "More transparent in what we want". One benefit relating to overview and transparency was the benefit of collocation [2]. James Dev described it cheerfully: "It is great that we can all sit in the same room and do this. The collocation makes you understand so much more".

Coordination and cooperation

The second most common answers to the open-ended questions were variants of "better cooperation/coordination/consensus" [8]. Several answers also highlighted how "improved communication" [3] contributed to better coordination and cooperation.

From discussions and interviews, respondents emphasized the benefits of feature ownership across teams [3], which Bank invented. One example given in an interview with Jocelyn Dev was: "We have often gained ownership of a feature, and then we are in charge of following up even those who are outside the team, and it has worked well. We are quick to make demands and follow up on them. It's a smart setup".

Another benefit presented in interviews was team stability [2]. Before, teams were organized in development teams and operations teams. Now, Bank experienced the benefits of teams with people from both development and operations:

> [When we heard that] people from operations would be part of our former team, I thought it would be difficult. It was not. In a short time, we have become very stable. We notice that we complement each other and that everyone contributes. (James Dev)

Speed, focus, and efficiency

In the open-ended questions, one answer mentioned "increased speed" [1] with the added routines. Another mentioned "focus" [1], and yet another "productivity" [1].

Speed and focus were not only related to producing results but also related to planning as such. Respondents said that they quickly perceived increased planning speed due to frequent PI planning events [3] in the first PIs.

> At first, I thought it would be just chaos, but after only a few times [of PI planning], it all "flowed" very well. Maybe because we had it so often, in the beginning, I think. (Jocelyn Dev)

Respondents also addressed how the PI planning routine made them finish planning on time [3], as in the quote by James Dev: "Although we

are pressured and feel a lack of time, this way [all teams planning at the same time] makes it possible to get the planning done on time".

Employee motivation

One answer to the open-ended questions simply expressed that working now with the added routines, was "great fun!" [1]. Respondents expressed more nuanced answers relating to motivation, such as how involving teams before PI planning [2] made a difference.

Jocelyn Dev explained that "we don't just go to planning unprepared. Instead, we have been given a chance to see and refine features before dividing the job [between teams]". Being part of the refinement process before the PI planning event was perceived as motivating for employees.

Also, respondents talked about improved motivation due to knowledge-sharing [3]. James Dev expressed the following: "I notice that many people share more than they did before. Showing their own solutions and stuff".

Planning precision

As presented in the three described PI planning events, planned work versus achieved results increased along the way, in almost every PI. This was confirmed when coding statements of progress reports were presented during PI planning events [6].

From having only 60 percent of the planned features done after PI 0, the percentage was continuously improving and reached 87 percent in the final two observed PIs, PI 4 and PI 5.

The improved planning precision was also highlighted by two of the respondents, Andrew Coach and Peter SM. They both expressed that it was becoming easier to promise managers what the train was able to deliver [2]. Peter SM expressed: "I think if you look at it from so-called higher management perspectives, they see the benefits of better promises [on what we are able to deliver] quite clearly".

Less dependent on individuals

Inter-team coordination routines are often intended for specific roles, at least as prescribed by the Agile frameworks. The Scrum of Scrums, for example, is according to SAFe intended only for the Release Train Engineer and the Scrum Masters or another representative of the team (Scaled Agile, 2020). Three respondents at Bank saw benefits of many roles attending meetings [4], not just the prescribed ones.

> From the beginning, it was just Scrum Masters and Product Owners [attending Scrum of Scrums], but then it turned out that this was in everyone's interest, to hear a little about how things were going and risks... And often, when something was discussed, there was like "Yes, but now we would have needed Charles here who knows more about this"? So the call to this meeting has gone wider and wider, and there is a lot of commitment and interest around it [the Scrum of Scrums]. (Annabelle RTE)

Respondents perceived an improved knowledge dissemination in the teams. This gave them an improved sense of motivation (as described above) but it also made team members less dependent on individuals [3]. Less dependency on individuals, in turn, might make team members more autonomous in their work.

This was also a goal of the implementation, according to Peter SM: "We aimed at becoming less dependent on individuals, and that is something we have achieved".

8.6.2 Perceived drawbacks

The thematic map of codes and themes of perceived drawbacks can be seen in Table 24, organized in the same manner as the perceived benefits above.

Table 24. Codes and themes of perceived drawbacks at Bank.

Codes from open-ended questions	**Codes from observations and interviews**	**Themes**
Too many meetings/overhead [11]	Too much refinement before PI planning [3]	*Too much time spent in meetings*
Inefficiency [3]	Big room meetings too long [3]	
Not Agile [2]	Long time between each release [3]	*Not tailored to the work process*
Not wholeheartedly implemented [1]	Last-minute changes before PI planning [3]	
Release difficulties [1]		
Coordinate details between teams [2]	Not enough overview [5]	*Less ad-hoc coordination*
Lack of communication/information [2]		
Unclear responsibilities [2]	Unclear dependency ownership [5]	*Unclear responsibilities*
Unclear roles [1]	Being a Product Owner for real [4]	
	Challenge for middle managers [2]	
Only our department works like this [2]	Coordinating different projects [3]	*Clashes with surrounding organizations*
Teams micro-managed [1]	Demanding refined features in advance [5]	*Limited autonomy*
	No time for reflection [5]	
Hard to see the individual contribution [1]	Not measuring added value [6]	*Not recognizing individual contributions*
Exclusion [1]	Working under the radar [5]	*Risk of exclusion*

The themes created for perceived drawbacks are explained below. Numbers in brackets show the number of answers from open-ended questions, observations, and interviews that belong to the same code.

Too much time spent in meetings

The, by far, most mentioned drawback in the answers to the open-ended questions was about having to attend too many meetings [11]. Also, some saw the time spent in meetings as signs of inefficiency [3]: "PI planning meetings become inefficient because of all latency".

Respondents perceived inefficiency, as the *Big room planning* meetings took too long [3] because features were not refined enough before PI planning. Elmer Test expressed: "We sit in those big room meetings way too long for it to be worth it. Could well be shortened if we did more

work in advance". This problem was also reflected upon during planning of PI 2:

> Once again, the feature breakdown was made too late. The team got to see these too late, and some people hardly know anything about [the features]. Today, POs are more like proxies; they do not own and control their backlog. The main problem is that there is rarely any pronounced benefit to the features. No epic owners who are passionate about certain kinds of features and can explain why they are needed. (Andrew Coach)

On the other hand, other respondents claimed the opposite, that too much time was spent on refinement before PI planning [3]. James Dev complained about this during an interview: "We have so many preparatory meetings. They never seem to be satisfied with how much we refine before PI planning".

Not tailored to the work process

Two answers regarding perceived drawbacks from the open-ended questions were that the routines seemed to be "not Agile" [2]. Bank implemented Agile ways of working several years ago, and the two employees perceived that the added routines were rather a step back. In one answer, the informant felt that the new routines were "not wholeheartedly implemented" [1], and another perceived "release difficulties" [1] due to too little tailoring.

Respondents in interviews also mentioned the problem of having too much time between each release [3]. This quote from Andrew Coach exemplifies how the routines were not tailored enough to deliver value often enough: "[There are teams] ...that get things done much faster, but I think the average time between each release is still 70-80 days".

Other respondents talked about difficulties in planning for such a long period because of the reccurring last-minute changes before PI planning [3]. This happened frequently, and Jocelyn Dev explained the problem: "[We] often have too big changes that sometimes occur before the planning meeting. Sometimes the same morning, we start planning, and it is not effective, but that is how it works here".

Less ad-hoc coordination

Another drawback mentioned in the answers to the open-ended questions was the difficulty of "coordinating details between teams" [2]. Two answers also suggested that there was a "lack of communication/information" [2].

In interviews, respondents talked about employees not having enough overview [5]. According to respondents, employees did not sort out coordination problems in an ad-hoc manner until much too late, when they perceived the actual damage. This problem did not only occur between teams but also between teams and stakeholders:

> They have sent out e-mails here to a lot of stakeholders and said that "right now we will have to focus on this and thus not this". These e-mails have been difficult to understand since they had not displayed a complete picture of what they have in their pipeline and should have been discussed instead. (Andrew Coach)

Unclear responsibilities

Two answers from the open-ended questions showed that one perceived drawback was "unclear responsibilities" [2].

During observations and interviews, respondents talked about the problems of unclear dependency ownership [5]. Elmer Test expressed how it could sometimes be difficult to know who should take the initiative: "[It is] often difficult to know who owns the issue. Especially when the dependency is on someone in another department. Like, should they or we push forward?".

This problem resides between teams at Bank within the same Agile Release Train as well. The Agile coach reflected on this during an interview:

> Then you might want to [in Scrum of Scrums] get even more initiative there, but we probably won't really make that happen I'm afraid. We would like teams to say, "but we might be able to help." Instead, so far, they are more like "this is our stuff". (Andrew Coach)

The Agile coach, as well as other respondents, also expressed how employees were not Product Owners "for real" [4]. Andrew Coach said that: "POs spend far too little time on being POs today, some only 50 percent". Developers also talked about the problem with Product Owners in the same way.

> They simply do not take enough time to be PO. Sometimes it feels as if they haven't realized it's their main job right now. That they are not technically involved as much as they need to make good decisions. (Jocelyn Dev)

Respondents also discussed that there was a clash between new and old roles, and especially a challenge for middle managers [2] to find their role. This quote from Peter SM exemplifies the challenge: "I think, if you look at it from a so-called higher management perspective, they see the

benefits of transparency quite clearly. But the middle managers are very difficult to get along with".

Clashes with surrounding organizations

One drawback mentioned in the answers to the open-ended questions was that "only our department works like this" [2]. As was explained previously in the Auto case chapter, this means that other departments had other routines that differ from Bank when it comes to inter-team coordination.

In interviews and observations, respondents discussed the difficulty of coordinating between different projects [3]:

> [It is] difficult when you need to synch with other projects. They don't work in the same way. So, most of what we do is in the same area, but some overall projects in the bank need certain parts, and they often want to interfere and decide on details, just like they did before. (James Dev)

Limited autonomy

One answer to the open-ended questions expressed a perceived feeling that "teams are micro-managed" [1].

However, in interviews and discussions during observations, respondents expressed limitations to autonomy due to the demands of refining features in advance [5].

Elmer Test said that "we can decide very much on our own, but then there is the issue of breaking down [features] a lot before planning". Jocelyn Dev concurred with this statement by announcing: "We are forced into more refinement now than we were before".

Another perceived limitation to autonomy, expressed especially in the beginning of the implementation, was not being allowed time for reflection [5]. A developer expressed the reason for this in the following way:

> Management has been bad at letting us have retros for the entire PI. It seems that they think it is a waste of time or something even though improvement along the way is the whole thing with Agile as I see it. (Jocelyn Dev)

When observing PI planning for PI 2, Andrew Coach expressed during a short break that: "We should have had an Inspect & Adapt [meeting] according to SAFe, but we didn't have enough time for it".

This confirmed the statement of Jocelyn Dev in that the Agile coach claimed that although they were supposed to have a meeting for reflection it was not prioritized by the managers.

Not recognizing individual contributions

In one of the answers to the open-ended questions, the informants also said that they did not belong to a team and, therefore, did not answer that part of the survey. One could, therefore, suspect that the respondent was a manager or an attending stakeholder. The drawback mentioned was that it was "difficult to see the individual contribution" [1].

The possible drawbacks of not measuring added value [6] due to the implemented inter-team coordination routines were mentioned several times during interviews and observations. In an interview with the Agile coach, he expressed this possible drawback based on the management of the program board:

> They wanted to remove the notes instead [when the dependency was solved]. Then I said, "yes, though you have no history if you do it like that". But they still did it. But without that [history], I would say that they solve it purely in terms of dialogue. They know when a dependency is solved. The team knows about it, but it doesn't show anywhere. (Andrew Coach)

Risk of exclusion

Another answer, only mentioned once in reply to the open-ended questions, was the drawback of possible "risk of exclusion" [1]. The informant suggested that there was a risk of exclusion if you, as an employee, was not positive to the ideas of large-scale transformation.

This risk was emphasized in several interviews where respondents explained that much of the past Agile work had been done under the radar [5]. A Scrum Master described what it looked like before the transition to Agile ways of working:

> We did skunk work. They [management] were terribly impressed with the speed and the knowledge [of Agile ways of working], which existed on the IT side of the business. And it really became the starting point for us to start working with these Agile methods, without having to go under the radar any longer. And that was shortly before the bank then started the big transformation. (Peter SM)

According to a developer, this 'skunk work' was, to some extent, evident even during implementation of the new inter-team coordination routines:

> Now there are more things that are out in the open, but since it is often required to follow exact routines, some have darkened a lot because they were afraid of the consequences. Under the radar, so to say. It would have been better if we could have talked more openly about what we were doing earlier. (Jocelyn Dev)

8.7 Changes to team autonomy

This section presents changes to team autonomy as they are perceived by employees at Bank. The perceived changes to team autonomy are analyzed based on whether they support or conflict with the Agile principles (Beck et al., 2001).

8.7.1 Too much team dedication

Peter SM expressed a sense of increased team autonomy at Bank but also said that it might come with a risk. According to him, teams must not forget to raise their heads to see the overall picture:

> Some teams have a tendency to try to isolate the team. You have to understand that you are really part of a larger whole and that the overall goal is more important than the team's own goal. […] it may be good with a certain competitive spirit […], but it should not mean that you start to protect your own team, to reach the team's goals instead of helping another team. It's easy to say, "we did what we promised". But maybe we should have helped the other team instead and lagged behind because it was more important than what your team did. You have to look at the overall goal all the time. (Peter SM)

The problem of isolating the team is both that more important work might be neglected, and work not needed might be performed. This behavior of team isolation in some teams at Bank is, therefore, not in accordance with AP 10 (*Simplicity - the art of maximizing the amount of work not done – is essential*).

8.7.2 Understanding the bigger picture

Catherine Dev explained that she perceived an increase in team autonomy since decisions in the team were now made at a higher level, and that understanding the bigger picture made the teams competent enough to make long-term decisions.

She said that this afforded a sense of being part of something bigger than your own contribution.

> What has worked well is that […] the knowledge of the whole business has definitely increased. You understand why you do things [...] that you are part of a larger whole. (Catherine Dev)

This change in team autonomy is in accordance with AP 8 (*Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely*). A better understanding of the

whole business makes it easier to plan and have the right expectations on different roles.

8.7.3 Architectural limitations

One of the Agile principles state the importance of design and architecture. In one quote, Philip Arch suggested that Bank did not make enough effort to recreate the technical architecture which had a bearing on team autonomy.

> But we are not really working on recreating the architecture […] to get an autonomous architecture where you can have full ownership of your product. (Philip Arch)

Philip Arch claimed that technical architecture poses an important limitation for team autonomy. Even if teams are organized according to value streams instead of systems as before, the architecture still limits how much the team can decide on, according to Philip Arch. This example shows limitations to team autonomy at Bank that are not supporting AP 9 (*Continuous attention to technical excellence and good design enhances agility*).

8.7.4 Less personal autonomy

During interviews and discussions, several informants mentioned that team autonomy had increased. However, some seemed to have experienced less *personal* autonomy.

> What has been quite tough for many, I think, is the change to be less isolated and maybe have to listen to things in meetings that may not be interesting to me today … have you been allowed to be a fairly isolated developer, then you may have quite a hard time having to listen to everyone else all the time too. Many [developers] have a hard time with it that you always have to share what you do. The value in it, getting everyone to understand the value of being transparent and listening to others all the time, is something you have to struggle with. (Peter SM)

Although there could be a personal price to transparency, as shown in this quote. Many developers, like Peter SM, reported that they mainly saw benefits of transparency. However, this example does show that the developers to some degree feel forced to participate in meetings, which is not in accordance with AP 5 (*Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done*).

9 Case Gov

The order and analysis of the case sections follow the research strategy presented in chapter 5 (Research Methods). As in the previous two chapters, the goal is to highlight findings that are particular to this case.

Gov (pseudonym) is a service development department in one of the largest government agencies in Sweden, and all people working at the observed department are native Swedish speakers. Hence, all presentation slides, discussions, and interviews were performed in Swedish, which means that quotes and texts presented in this section have been translated into English. Just as with Case Auto and Case Bank, all respondents are anonymized and have been given other names. Instead of family names, abbreviations of their roles are used. Two new abbreviations are used in this case compared to Case Auto and Case Bank. The abbreviation *OpDev* is short for the role of Operations Developer. An Operations Developer is someone who develops services for citizens in the government agency. A similar role in a corporation is called a Business Developer. The second abbreviation is *Epic*, which is short for the role Epic Owner suggested in SAFe (Scaled Agile, 2020). Table 25 presents the interview respondents and employees quoted at Gov.

Table 25. Quoted employees at Gov.

Alias in quotes	**Role at Gov**	**Interview date**
Alina Coach	Agile coach (consultant)	2018-01-10
Helena RTE	Release Train Engineer	2018-01-10
Pierre SM	Scrum Master	2018-03-02
Cecilia OpDev	Operations Developer	2018-03-02
Andy SM	Scrum Master	2018-06-01
Carl Dev	Software Developer	2018-06-01
Jean Mgr	Department Manager	Quotes from observations
Andrea Epic/RTE	First as Epic owner, later as Release Train Engineer	Quotes from observations
Olivia Dev	Software Developer	Quotes from observations

The government agency has offices in several cities in Sweden, and the studied department, Gov, consists of teams from three different sites. Agile ways of working were introduced already in 2012, but Gov did not start implementing inter-team coordination routines until the spring of 2017. Gov decided to implement routines prescribed by SAFe. Gov is organized into one Agile Release Train with one Scrum Master and one Product Owner per team. While setting up the train, the teams were, to

some extent, reorganized. They originally only contained Software Developers and testers, but when the train was formed, Operations Developers were added to the teams from another department.

The implementation of SAFe at this department was intended as a pilot project, as the first implementation at the government agency. The idea was to learn from this implementation to construct a template and guidelines for other departments. The pre-study documents state several intended goals and expected results from implementing inter-team coordination routines from SAFe:

- Consensus on goals and direction.
- Better overview and increased transparency.
- Even workload for the teams.

Figure 17 shows a timeline of the performed observations at Gov.

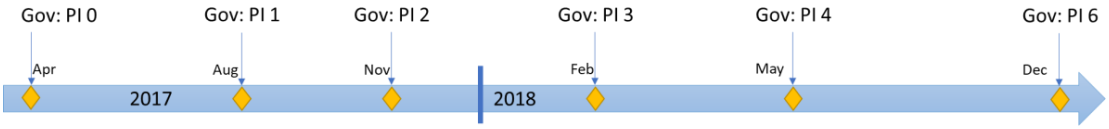

Figure 17. Timeline of PI planning events when data was collected at Gov.

PI 0 in April was not a real PI planning event with the teams in the train but rather a one-day planning occasion for managers and stakeholders. The following text was sent out in advance to explain the purpose of the PI 0 planning event: "Plan the work that needs to be done before we can start activities in the Agile Release Train … Gather all the managers within the program to create the conditions for a successful start together".

Therefore, the first act presents the planning and execution of PI 1, the second act PI 3, and the third act PI 6.

9.1 The First Act: PI planning for PI 1

The main Agile coach, Alina Coach, acted as the first Release Train Engineer and was responsible for planning and moderating the first PI planning at Gov. From planning PI 0 up until this event, managers organized in the so-called *Core Team* decided what roles and teams should be part of the Agile Release Train. Initially, they decided that four teams were to be part of the train, but during PI 0, an additional four teams were added.

> The Agile Release Train organization was initially designed by the managers of the business department and the IT department. The proposal was received with mixed enthusiasm from the employees. (Alina Coach)

When starting the PI planning on the August 29, 2017, the department manager at Gov, Jean Mgr, explained that only six out of the eight teams would be present to conduct the PI planning. Two teams had to remain in their hometown, and instead, they attended virtually using Skype and cameras in their offices. The manager showed two slides to explain the reason for implementing large-scale Agile ways of working.

> - Governance of the agency's IT development unsatisfactory, "Gov initiative 2016."
> - We need to review our work methods and forms of governance. Gov department is conducting a pilot project. The pilot project also covers operations development.
> - Responsible for operating the business efficiently, by the law, and to manage the state's funds accordingly, "Government regulation".
> - The goal is to establish an Agile ways of working based on the SAFe framework
> - The pilot should be indicative and will support and simplify a step-by-step introduction throughout the rest of the organization
> - Experiment! We develop and evolve while carrying out the pilot project.
> (Presentation slides from PI planning event for PI 1)

After this presentation, a team-building coach from another department took the floor and presented a theory about team development and lead the teams in a team-building exercise. A leadership coach from yet another department explained that she would assist the managers during the first real PI. Alina Coach, who was a hired consultant, presented four internal Agile coaches from other departments in the government agency. These four Agile coaches would assist the teams during the first three PIs. All in all, seven coaches were present during this PI planning event.

> This is just typically [Gov], instead of managers driving the change, they appoint change managers and coaches to do the job. (Carl Dev)

Alina Coach continued to explain the reason for conducting the PI planning routine and that she would have the role of Release Train Engineer. According to her presentation slide, the reason for conducting PI planning was to:

> Get consensus within our team of teams, access to common resources (such as architects and UX designers) during planning, clarify ambiguities directly and effective communication. (Slides from presentation)

When it was time to start planning in the team breakout session, Alina Coach explained that they would plan for four sprints, since that is what SAFe prescribes. With three-week sprints, this meant a planning horizon of 12 weeks. She then presented the roadmap, extending over three future PIs described in great detail. This had been put together by the managers in the Core Team. Alina Coach commented on this roadmap in a short discussion during lunch on the first day.

> This first version of the roadmap is very detailed. The roadmap will need to be reworked for the next PI planning event, as it is not intended to be at such a detailed level. (Alina Coach)

Although 15 hours were used for the first PI planning at Gov, only 4.5 hours were spent in team breakouts: the last two hours of the first day and the first two and a half hours of the second day. During the first day, several employees shouted for assistance from the Agile coaches during these two hours and complained that they did not know exactly what they were supposed to do.

The teams attending virtually via Skype complained about too much administration. This was due to the Agile coaches telling them to do everything by the book, according to SAFe. This entailed doing all planning on paper post-it notes, but since they were attending via Skype, they still needed to transfer their results to a digital environment to be able to present their results to the rest of the teams.

At the end of the first day, Gov conducted a "Management Review & Problem Solving" meeting as prescribed by SAFe. Managers, Product Owners, Scrum Masters, and several stakeholders not belonging to a team gathered, which made it a total of 35 attendees in this meeting. Alina Coach stated what she saw as the main problem thus far.

> There are many who take great responsibility for delivering different parts, but few feel responsible for delivering a full solution. That is an approach that needs to be worked into more roles. (Alina Coach)

Several of the Scrum Masters pointed out that their teams thought that the planning horizon was too far into the future. They complained about not being able to foresee all dependencies and coordination needs for such a long period. Managers told them to do their best. The Operations Developers complained that there was too much focus on the specific details of IT development and not enough on actual functionality and business value.

The next day, after team breakouts, each team presented their plans. Several Scrum Masters also thanked the Agile coaches who had helped them to create these plans. The Release Train Engineer Alina Coach commented that she was worried about all plans being so crammed full with no room for additional work. One of the managers bluntly commented that this was how they normally planned.

9.1.1 Virtual Scrum of Scrums once a week

The Scrum of Scrums was held each Wednesday for 30 minutes, between 10.30 and 11 o'clock. Since teams worked in different cities, all meetings were virtual using Skype for Business. The meeting always followed the same agenda:

1. Information from the Release Train Engineer
2. Information from the Architect
3. Information from Product Management
4. Information from the Test Manager
5. Information from teams that is of interest to other teams (new dependencies etcetera)
6. General risks/obstacles/problems

This agenda was set by the Core Team and did not follow the prescribed format in SAFe. Instead, the agenda was to some extent similar to the weekly status meeting agenda in the prescribed project management model used at Gov. SAFe prescribes four items, similar to the agenda of the Daily Scrums (Scaled Agile, 2020), focusing mainly on what has been done, what will be done, and obstacles. The Gov agenda is instead focused on information-sharing.

At Gov, attendance was mandatory for Scrum Masters at these meetings, and if a Scrum Master could not attend, he or she appointed another team member to attend. Apart from the Release Train Engineer, at least one person from product management and maintenance was obligated to attend, while Product Owners were not invited.

9.1.2 Unclear and changing roles during the PI

Former Project Managers, now converted to Epic Owners, which is a role prescribed by SAFe, complained that they did not understand what was expected of them in terms of coordination. A manager from the Core Team expressed that the difference between Epic Owners and Project Managers would be further investigated together with the Agile coaches

and clarified later. One of the previous Project Managers, Andrea Epic/RTE, expressed her frustration during an interview.

> They say that I should be an Epic Owner now, but still, no one has approved that we should not follow our project model. Does that mean that I need to both do things that Epic Owners should do and still do everything as a project manager? I don't get it. They say I shouldn't need to do all the project management stuff, but they don't know who should do it instead. (Andrea Epic/RTE)

Two weeks into the PI, the Core Team decided to let the employee Elena RTE, formerly a Project Manager, take on the role of Release Train Engineer at Gov. Since Alina Coach was a consultant, it was deemed more appropriate to let a government employee have this role.

This came as a surprise to Elena RTE, who had only attended a basic training course about SAFe and had no prior experience of working according to SAFe. This was decided on a Wednesday morning, which meant that the first job for Elena RTE was to lead the second Scrum of Scrums meeting at Gov.

9.1.3 Few dependencies between teams in the train

Alina Coach showed the program board during final presentations of PI planning, see Figure 18. All rows above the orange line, starting with a blue marking of the first column, are teams within Gov. Everything below, starting with a pink marking, are teams in other departments. The red strings shows dependencies between features owned by different teams.

Alina Coach concluded that there were few dependencies between teams within the train and many more to teams in other departments.

Figure 18. Program board showing dependencies at Gov.

Although the program board looks like it contains many dependencies, Gov realized that teams had different views on what a dependency was. Some defined dependencies as "when a team needs something delivered from another team," while others defined it as "when a team needs information about a feature from another team." The later definition turned out to be the most common.

The Core Team decided *not* to use this as a correct definition and therefore removed all dependencies from the program board that were not "when a team needs something delivered from another team". After this decision, only three dependencies remained within the train and eleven dependencies on teams in other departments.

9.1.4 No IP sprint but half a day for PI retrospective

Managers at Gov did not want to allow time for an IP sprint. Instead, PI 1 ended with a half-day of retrospective meeting, from 1 pm to 5 pm, for the whole PI period with the whole train. This was followed by two days of PI planning for the next PI.

At the beginning of the retrospective meeting, Elena RTE presented some measurements for the train. All objectives formulated in the teams had been given a business value, and the total sum of these planned values was 206 points.

The amount of business value achieved, according to the Core Team, was 172 points, which meant a fulfillment rate of 84 percent.

9.2 The Second Act: PI planning for PI 3

PI 2 was extended to five sprints instead of four. The reason for this was that Christmas and New year took place during the PI, and many employees took some vacation. With less available man-hours, the Core Team decided to extend the PI so that roughly equal amounts of planned features could be prepared and suggested in PI 2. For PI 3, they were going to plan for four sprints once again.

Since the Core Team wanted the teams to conduct a system demo and with teams in different places, an additional half-day (afternoon) before PI planning was used for the demo with the different teams. This meant that three teams traveling from other parts of Sweden had to stay for three days, starting on February 13, 2018.

During the systems demo, which concluded PI 2, the teams expressed a lot of frustration.

> We are making changes to follow what is prescribed in SAFe, but that does not help the team going forward. (Pierre SM)

Several of the teams had not reached their objectives, and one team member (Carl Dev) said that there had been much work added during the PI from product management without any work being taken out. Another team member (Olivia Dev) expressed that although they were supposed to work full time in their teams with backlog items that had been planned for in PI planning, team members were taken out for other kinds of work. In practice, that meant that managers ignored what had been decided in PI planning and instead gave employees other tasks during the PI.

> [Employees in] the teams are now fully allocated to the train, but then there is reality. Managers think they can always prioritize other work than what was planned for. Always. (Olivia Dev)

For the planning of the third PI, two teams were not allowed to attend physically. The reason was that managers decided that the teams needed to focus on a software release that would take place during the planning days. Instead of attending online using Skype as when planning PI 1, the teams had conducted their plans the week before. Their plans were supposed to be integrated into the overall PI plans during the PI planning event.

During this PI planning, other reasons for conducting PI planning were presented in the initial presentation slides: "We benefit from meeting each other! We get a common picture of what has been done, what challenges we have, and where we are heading. We will get to know each other and promote teamwork."

A woman from the department responsible for service development presented 'The customer journey' with process charts showing what kind of systems a citizen in need of service from the agency actually faces. After the presentation, a Software Developer expressed how important this presentation was for him since it gave such a good overview and understanding of what his team was doing.

Elena RTE presented some reminders of what the teams should think about during team breakout. She highlighted that the four Agile coaches were present and that any team in need of assistance could receive help. One of the Agile coaches stressed the importance of updating the program board continuously. Another Agile coach stressed that, since several teams did not reach their objectives during the last PI, they needed to plan for no more than 85 percent of the team's capacity.

During the systems demo, several employees made frustrated comments during team breakouts. A Scrum Master commented to a Product Owner that they were handed their features, instead of being able to choose for themselves. She then asked why they had not been able to look at them in advance since they were already intended for her team. The Product Owner only answered that he did not know why.

During team breakouts, the Core Team had a meeting with the Agile coaches and Elena RTE to discuss experiences of the PI planning event so far. Alina Coach expressed her worries regarding too much work planned for each sprint in the teams.

> But there is still too much PUSH, no PULL! Still, resource optimization. Everyone plans for 100 percent [of available capacity] all the time! (Alina Coach)

One Agile coach suggested that it would be a good idea to start-up Communities of Practice, since several Scrum Masters had expressed the need for learning from each other. The suggestion was immediately criticized by the department manager, Jean Mgr.

> No, we can't allow that. That would take away too much of their productive time. (Jean Mgr)

Alina Coach responded to Jean Mgr's comment that this was an example of their resource optimization approach. Jean Mgr ended the meeting and acknowledged that they probably still made too many decisions based on old ways of thinking. However, she did not change the decision regarding not letting Gov start Communities of Practice.

One developer asked his Product Owner when they would be allowed to stop doing things expected from the project management model, since they had implemented new routines from SAFe. The Product Owner answered that the Core Team did not have any further information about that. A tester claimed that what the Agile coaches taught was not how things were done in reality. She claimed that the Agile coaches were competing regarding who knew the details of SAFe best, rather than helping teams tailor routines according to their needs. Another developer tried to convince the team that they should not plan for too much, as the Agile coach had stressed not planning for more than 85 percent of their capacity. The Scrum Master of this team argued that it was better to plan for full capacity and explained why in the following way:

> My experience tells me that it is not a good idea to plan for too little, even though we say 85 percent today. In another team I worked in, we even planned for more than our capacity and still succeeded in meeting our objectives. (Andy SM)

This time, the teams were allowed 6.5 hours of team breakout time, but several developers expressed still feeling stressed and not having enough time to plan. At the final presentations of the planning days, much of the frustration had settled. Several developers claimed that they had a much better overview of the other teams' work. An Operations Developer expressed that it seemed like the managers were finally letting them, the employees, be a part of the change, that they were finally listening to the "people on the factory floor". Alina Coach commented this in an informal meeting with the Core Team during a coffee break at the end of the PI planning event.

> There are changes in the organization that employees are more actively involved in now. We have started to move towards more autonomy in the teams. One effect of this is that we do not encounter the same resistance as we did when employees were not involved in the beginning. (Alina Coach)

9.2.1 Even fewer dependencies within and outside the train

Having decided on a common definition of dependency (*when a team needs something delivered from another team*), only one dependency was identified

within the Agile Release Train during PI planning. Nine dependencies were towards teams in other departments.

9.2.2 Splitting the Agile Release Train in two

During PI 3, the Core Team decided to split the Agile Release Train into two trains: X-train and Y-train. The X-train was put together with three teams working in the same office in a city in northern Sweden. The Y-train consisted of one team from the same northern city and two teams located in a city in southern Sweden. The two teams from the south of Sweden used to be three teams during PI 2, but now, two of them were mashed together into one team. The final team, located in a city in central Sweden, consisted mainly of Operations Developers. This team was not put in any of the trains, but instead, they would assist both trains as a *shared service* team. The operations development team had, therefore, their own separate backlog.

This splitting of the train meant separating the original program backlog into three separate backlogs: X-train backlog, Y-train backlog, and Operations Development team backlog. It did not, however, change the way Scrum of Scrums was performed where all teams from both trains still attended.

Elena RTE remained as a Release Train Engineer for the Y-train while Andrea Epic/RTE, also a former Project Manager, was now appointed as a Release Train Engineer for the X-train instead of remaining an Epic Owner. Andrea Epic/RTE did not have any prior experience of working according to SAFe and had, just like Elena RTE, only attended a basic training course on SAFe.

9.2.3 Few completed objectives compared to plans

Instead of only measuring achieved business value, the Core Team decided to measure the number of objectives and features being completed, on-going, or not started. For the X-train, Y-train, and the operations development team, the fulfillment rate of business value was deemed to be 91 percent, 78 percent, and 74 percent, respectively. However, looking at completed objectives, they were, respectively, 11 out of 17, 3 out of 8, and 0 out of 6. That meant that in total only 14 out of 31 objectives were met (45.1 percent).

9.3 The Third Act: PI planning for PI 6

Although the train was divided into two during PI 3, both trains decided to conduct PI planning on the same days but located in different places. The X-train, where all teams resided in the same northern city in Sweden, conducted the PI planning in their own office while the teams of the Y-train met up with the operations development team and the Core Team in the city in central Sweden. The Core Team decided to only spend two days on PI retrospective, systems demo and PI planning and started on the morning of Tuesday, December 4, 2018.

The PI retrospective was conducted separately at the different locations, but at 10.30, the X-train joined via Skype to participate in systems demos and to receive information from the Core Team. Jean Mgr presented objectives and the overall business plan for the coming period. Jean Mgr also presented additional reasons for implementing SAFe: "To better focus on 'those we are here for' at work, to give product management and the teams a clearer prioritization".

However, as Jean Mgr mentioned, since the budgeting process for 2019 had not been finalized, the plan of operations could not be finalized either, not even the first full quarter of 2019. This could mean that all teams would not be able to plan for all four sprints in the PI. The PI should, however, still consist of four sprints, according to Jean Mgr.

Elena RTE and Andrea Epic/RTE, the two Release Train Engineers, presented practical information about the planning event. None of the Agile coaches were present, and Andrea Epic/RTE informed that Gov was not allowed help from coaches anymore. Elena RTE explained that they had decided to stop conducting some of the routines, since there were no Agile coaches there anymore to tell them what they must do.

> We can work more freely now when we don't have any "method people" around. (Elena RTE)

With those words, Elena RTE received scattered applause around the room. Someone made a low cheer. Elena RTE also described that they would not use the program board to show dependencies anymore. Instead, if a team was responsible for a feature with a dependency on another team, the team should write this as a comment in the feature information in the backlog.

During this PI planning event, teams could plan more for themselves in team breakouts and listen to fewer presentations. Out of 14 hours set aside for the PI planning event, eight hours were spent in team breakouts.

The PI planning ended with a one-hour retrospective meeting about the PI planning event itself. Several team members expressed that eight hours was too much time for planning in breakout sessions, especially since some of the teams could not plan for the final fourth sprint of the PI. One team could not even plan for the final two sprints, due to the unfinished budget of 2019. Several team members in the Y-train also questioned the need for traveling and sitting in the same office. They thought they would be able to make their plan just as well without meeting the other teams physically. Since they almost had as many discussions with the teams in the X-train via Skype as with teams in their own train, they thought they could have used Skype in discussions with all of them.

9.3.1 Abandoning the program board

Since the program board was abandoned during PI planning, the routine of updating dependencies visually on the program board was obviously discarded. Scrum of Scrums, on the other hand, was still held in the same way with the same agenda.

When new dependencies were identified and presented in Scrum of Scrums, the team responsible for a feature with a dependency updated information about this dependency in the product backlog together with other information about the feature. This meant that dependencies could only be identified by reading information about every single feature.

9.3.2 Each team responsible for the coordination of their own features

With information about dependencies only residing in the product backlog, each team was instead responsible for the coordination of its own features. The decision was made due to the possible disadvantage of not being able to get an overview of dependencies anymore. However, according to one team member, it might also have a positive effect. The team member expressed that it had become much easier to talk to and help other teams now that everyone felt more responsible for their own features.

9.4 Identified institutional logics

Table 26 summarizes the different dimensions that present the beliefs and values of specific institutional logics, creating an institutional logic ideal type consisting of Agile rulebook logic and resource efficiency logic.

Table 26. Institutional logic ideal type at Gov.

Dimension	Agile rulebook logic	Resource efficiency logic
Root Metaphor	Agile routines as rules	Efficiency as utilization of man-hours
Sources of Legitimacy	Unity in interpretations, standardization	Personal efficiency
Sources of Authority	Regulations and method descriptions	Management
Sources of Identity	Method knowledge	Clearly defined roles
Basis of Attention	Level of method adoption	Resource availability
Basis of Strategy	Adopt method of choice	Utilize resources

Five Agile coaches helped Gov implement inter-team coordination routines during the first five PIs. Four of the coaches worked at the government agency and had all been to the same SAFe training courses. Alina Coach was also trained in SAFe with experience from implementing it in another organization previously but did not have previous experience of working at Gov. Since SAFe was the baseline for all of them, the inter-team coordination routines were all implemented by the book. A tester claimed that the Agile coaches were competing with each other regarding who knew the details of SAFe best, rather than helping Gov in how to tailor routines according to their needs. This evidence suggests an *Agile rulebook logic* to be dominant at Gov.

One specific example of Agile rulebook logic was how planning was performed by teams attending virtually in the PI planning event. All planning was to be done on paper post-it notes as prescribed by SAFe but as they were attending via Skype, they still needed to transfer their results to a digital environment to be able to present their results to the rest of the teams when the final plan was presented.

Another example of the Agile rulebook logic that dominated at Gov was what happened during the delayed annual budget process during PI 6. Without a definite budget, managers were not allowed to plan for more work than a couple of months into the new year. Still, the decided PI length of four sprints was kept, since SAFe prescribes a planning period of four sprints (Scaled Agile, 2020). This meant that several teams were unable to plan for their final one or two sprints. The absence of tailoring while the Agile coaches still attended was evident at Gov. As the Agile coaches left before PI 6, however, Gov employees expressed that they now felt free to make changes to start tailoring the inter-team coordination routines. So they did, but mainly by stopping to use some of them.

This does not point to Agile toolbox logic, since the tailoring did not suggest implementing or inventing new routines, only discarding previously implemented routines. There are several examples of a preference for resource efficiency logic at Gov. Teams were not allowed an IP sprint, only a short retrospective meeting. Teams planned for full capacity, even if evidence showed that they would not be able to deliver everything they had planned for. Communities of Practice were not allowed to be started, since it would reduce productive time for the employees.

> But there is still too much PUSH, no PULL! Still resource optimization, everyone plans for 100 percent [of available capacity] all the time! (Alina Coach)

As can be seen in the quote above, Alina Coach even explicitly stated that there was too much focus on resource optimization, which is contrary to the intended logic of SAFe (Scaled Agile, 2020). Signs of resource efficiency logic could also be seen in a comment from Andy SM, who discarded the idea of only planning for 85 percent of capacity and preferred to plan for full capacity.

9.5 Analysis of coordination routines

In this section, different aspects of routines will be analyzed. Dependency types will be analyzed, and the identified performance and tailoring of routines will be presented. Table 27 summarizes differences between SAFe and how routines were performed at Gov.

Table 27. Routines at Gov from a tailoring perspective.

Routine	According to SAFe	Gov
PI planning	Set cadence, PI plan every 8-12 weeks.	Mostly 12 weeks. 15 weeks for PIs stretching over Christmas and summertime.
	4*2-week sprints and IP sprint (2 weeks).	4*3-week sprints, no IP sprint. 5 sprints for PIs stretching over Christmas and summertime.
	Duration: 2 days.	Duration: 2 days.
Scrum of Scrums	Twice weekly.	Once a week.
	Duration: 30 minutes and "meet after".	Duration: 30 min, no "meet after".
	Attendees: RTE, SMs, and possibly stakeholders.	Managers, RTE, SMs, and stakeholders.
Program board	Contains: Feature delivery dates, feature dependencies, and milestones.	Contains: Feature delivery dates, feature dependencies, and milestones.
	May, or may not, be maintained during PI.	Not maintained during PI. Later abandoned.

Looking at the PI planning routine, the first session was only allowed 4,5 hours of work but was later increased each PI, although the number of sprints was the same. Table 28 shows the time and intervals of the PI planning and the PI period.

Table 28. PI planning data from Gov at PI 1, PI 3, and PI 6.

	Gov PI 1	**Gov PI 3**	**Gov PI 6**
Number of weeks per PI	12	12	12
Number of sprints per PI	4	4	4
Sprint length (weeks)	3	3	3
Length of final Inspect and adapt meeting	None	1 hour	30 min
PI planning event (hours)	15	14	14
Team breakout time (hours)	4,5	6,5	8
Team break-out time (percent):	30 %	46.4%	57.1%

To summarize the tailoring conducted at Bank relative to prescriptions from SAFe (Scaled Agile, 2020):

PI Planning

- Cadence based on available man-hours instead of weeks
- An increasing amount of team breakout time
- Conducted virtually, teams in different places

Scrum of scrums

- Once a week
- Managers also attending
- Mostly information-sharing
- Conducted virtual, teams in different places

Use and management of the program board

- Only one type of dependencies

At Gov, the length of the PI was tailored based on the number of available man-hours. During PIs covering the summer months, where most people in Sweden are on holiday leave for a few weeks, PIs were longer to compensate for less available man-hours in the teams. A similar condition goes for Christmas and New Year's. What was important for a PI cadence at Gov was the number of man-hours to be planned for, not the calendar cadence. The department manager at Gov did not allow for an IP sprint, only a short (less than two hours) retrospective meeting and the two days for PI planning, following the standard agenda from SAFe.

The reason for not allowing a sprint for inspection and adaption was that it was considered a waste of time and money to spend so much time on non-productive activities. From the perspective of institutional logics, the legitimacy for this decision was based on a preference for resource optimization counter to the flow optimization intended in SAFe.

The standard agenda from SAFe was followed and performed in the same manner during all observed planning occasions with only minor changes. But team members complained that planning felt "non-Agile", since the workshop was conducted in the same way each time and that plans were not updated during sprints, making the whole PI seem like a commitment, instead of a plan with intended flexibility at the start of each new sprint.

The Scrum of Scrums was conducted once a week on a strict cadence: Wednesdays at 10.30 to 11.00. Apart from the Release Train Engineer and Scrum Master, several managers and stakeholders also attended. The meeting was always held virtually (via Skype) and anyone at Gov could listen in during the meetings.

According to SAFe, Scrum of Scrums is a meeting to help out between teams, not an information meeting. At Gov, managers and stakeholders participated to inform Scrum Masters about important decisions and status updates on areas surrounding the project. All parties at Gov viewed the Scrum of Scrums primarily as an arena for information sharing, and secondly, as a meeting for highlighting risks and dependencies. Only one of six agenda items highlighted dependencies. The rest aimed at information-sharing. The meeting had the exact same format every time.

During the first PI planning event at Gov, all milestones, features, and dependencies were presented on the program board. The board showed that all features had dependencies either with other teams or other departments. But there were different interpretations of what the term 'dependency' means. Scrum Masters and team members presented a dependency on the board when the team only needed information from one person outside the team, while the Release Train Engineer and stakeholders defined a dependency as a need for a specific systems feature to be fulfilled before the dependent function could be developed.

After a brief discussion at the end of the first PI planning event, all parties agreed on the latter definition, which only left three dependencies within the train and eleven dependencies to other departments. The program board was not updated during the PIs, and since so few dependencies between teams were recognized during PI planning, Gov abandoned

using the program board altogether during PI 6. Instead, teams recorded dependencies in their own team plans.

9.5.1 Dependency analysis

During the initial PI planning, the program board was filled up with lots of dependencies. When the Core Team started to question the dependencies, it became evident that there were different views on what a dependency was. Some defined dependencies as "when a team needs something delivered from another team", while others defined it as "in need of information about a feature from another team". The Core Team decided not to use the latter definition and therefore discarded all dependencies from the program board that did not fit the description "when a team needs something delivered from another team". From a coordination theory perspective, the Core Team had decided that a *producer-consumer dependency* was the proper way to define dependencies, and nothing else. By only putting these dependencies on the program board, they decided that the need for coordination was based on the two sub dependencies *transfer* and *precedence*.

By analyzing this in terms of workflow between actors based on Thompson's (1967) classification, the Core Team decided that only *sequential dependencies* were to be acknowledged. Although, as the first version of the PI planning showed, most of dependencies, according to the other definition, were, in fact, *pooled interdependencies*. This was also evident in discussions with team members, when they claimed that it caused a lot of waiting when there was not enough time to discuss issues with members of the operations development team. Unfortunately, these *pooled interdependencies* were not allowed on the program board and thus could not be tracked, although they were in the majority.

Analyzing static versus emerging dependency issues became complicated due to the problem expressed above. Since only *sequential dependencies* were defined, emerging dependencies of other types could not be identified. However, by analyzing the transcripts from nine Scrum of Scrums conducted during the first three PIs, it was possible to calculate the number of raised additional sequential dependencies during this period. No new dependency emerged during the first PI, while two new dependencies emerged during the second and two during the third PI – all within the train. So, only four new dependencies emerged during these three PIs. The number of identified dependencies during PI planning within the train for the same period was three for PI 1, nine for PI 2, and one for PI

3. That is a total of thirteen dependencies, which suggests that 23.5 percent were emerging dependencies during the first three PIs.

Although the sequential dependencies can be considered few, it shows that emerging dependencies were common and that coordination routines to handle them were of importance to Gov.

9.6 Perceived impacts

At the beginning of the PI planning event for PI 3, a paper questionnaire was handed out to all participants to be filled out. The full questionnaire can be seen in Appendix B.

Several parts of the questionnaire were inspired by Laanti et al. (2011), and the first section was intended to give an understanding of the employees' views of the different parts of SAFe. Statements with Likert scales where used, ranging from 1 to 5, where 3.0 can be considered neutral. The results can be seen in Table 29. At Gov, 65 answers were collected from survey respondents out of 70 employees (a response rate of 92.9 percent).

Table 29. Perceived impacts of inter-team coordination routines at Gov

Survey Question (Likert scale 1-5: 1=Not true, 5=Very true)	**Mean**	**SD**	**Median**	**Mode**
The PI planning gives me a good overview of our work	3.2	1.0	3	4
The Program Board is very helpful for coordination between teams	2.8	0.9	3	3
The Scrum of Scrums (SoS) meetings solve inter-team coordination problems	3.2	0.8	3	4
There are many problems with working in a large-scale setting as ours, using SAFe	3.9	0.8	4	4
I would like to go back to the old way of working (instead of working according to SAFe)	2.6	1.2	3	4

As Table 29 shows, the view of the PI planning session and Scrum of Scrums both averaged at 3.2, which shows a mildly positive view of its usefulness. The view of the program board averaged at 2.8, a mildly negative view.

The average for the fourth statement (*There are many problems with working in a large-scale setting like ours, using SAFe)* was 3.9, which shows a view of the implementation as very problematic. The average for the final statement (*I would like to go back to the old way of working (instead of working according to SAFe))* was 2.6. This shows that, although close to neutral (but with a

Mode of 4), many employees would like to go back and did not appreciate working according to the implemented inter-team coordination routines at Gov.

9.6.1 Perceived benefits

The defined themes can be seen in the right-hand column in the thematic map displayed in Table 30.

Table 30. Codes and themes of perceived benefits at Gov.

Codes from open-ended questions	**Codes from observations and interviews**	**Themes**
Coordination, cooperation, and sense of community [11]	Coordination [8]	*Coordination and cooperation*
Teamwork [2]	Easy to help each other between teams [8]	
Division of work [1]	Many coaches, more cooperation [4]	
Overview [7]	Overview [5]	*Overview and transparency*
Visibility/Transparency [5]	Understanding where we are heading [4]	
	Being able to demo a full feature [2]	
	Management makes better decisions. [3]	
Better prioritization [5]	Prioritizing everything, not just IT [3]	*Clear priorities*
Speed [1]	Focused due to longer plans [4]	*Speed, focus, and efficiency*
Work better prepared [1]	Deliveries more frequent [3]	
Efficiency [1]		
Competence dissemination [3]		*Competence dissemination*
	Understanding the client perspective [5]	*Understanding the client perspective*

Numbers in brackets show the number of answers from the open-ended questions that were somewhat similar, belonging to the same code. To give an example, the code "better prioritization" comes from answers such as "clear prioritization", "better explained priority", "product management priorities", and simply "priorities".

Coordination and cooperation

The most common answers to the open-ended questions were different accounts of improved "coordination, cooperation, and sense of commu-

nity" [11]. Two answers mentioned the "improved teamwork" [2] and the importance of "division of work" [1].

Records from observations, as well as interviews showed that coordination [8] had improved. Two things noted during reflection on PI 1 before PI planning for PI 2 started was: "Better collaboration/coordination between different development teams. Stakeholders and product management have greater visibility and are more accessible to everyone". This view was confirmed by Andy SM in an interview where he explicitly expressed the benefits of PI planning: "The coordination of teams is very good. Who does what and when becomes so clear when everyone is planning at the same time."

Four respondents talked about how it was much easier to help each other between teams [8]. An Agile coach expressed how she perceived this improvement:

> It is much easier to help out between the teams. I see how often they take the initiative to ask for help and get help from each other during those two days [of PI planning]. (Alina Coach)

Respondents claimed that one benefit of many coaches attending was better cooperation [4] between teams. Cecilia OpDev exemplified this: "It is much easier to help one another … And also to give each other help, where I think the coaches have been most helpful". Regardless of whether the coaches played a key part in this or not, the helping out between teams was also mentioned in Scrum of Scrums, where Pierre SM expressed his gratitude: "We have had tremendous help from each other and from our colleagues in the other teams".

Overview and transparency

The second most common answers to the open-ended questions were variants of "better overview" [7]. Also, several answers related to "visibility/transparency" [5].

In both observations and interviews, improved overview [5] was mentioned. Helena RTE claimed that the single most important benefit of the implemented routines was: "The overview. That we know what everyone is doing". In a discussion regarding what was positive during the planning of PI 1, the team members of Team 1 expressed: "We have been given a complete picture of all the tasks of the [Agile Release] Train".

Three respondents talked in interviews about better understanding of where Gov was heading [4], the long-term view. Andy SM explained that: "It is easier to make our own decisions in the team now that we know

where we are heading". The understanding was not only due to actions within the Agile Release Train but also, as presented by Helena RTE: "People both from operations and the customer side comes to present their views, and that gives us better understanding of the long-term objectives".

Two respondents also talked about the beneficial transparency in being able to demonstrate a full feature [2]. During the demonstration of results in the day before PI planning of PI 3, Carl Dev expressed: "It's great to see a real demo on feature level where the entire functionality was demonstrated live".

Two Scrum Masters and one coach expressed during interviews that the new transparent way of working made the management to make better decisions [3]. Andy SM said: "I notice that POs and managers have a great advantage of this [way of working]. They do not make decisions on such loose grounds but participate more and hear all discussions".

Clear priorities

From the open-ended questions, five answers mentioned the perceived benefits of "better prioritization" [5] with the added routines.

In interviews, respondents expressed in more detail that an added benefit was that they were now prioritizing everything, not just IT [3] work. One of the Scrum Masters explained what *everything* meant at Gov:

> One extraordinarily good thing about getting operations development people in our teams is that the priorities now include everything. Not only IT functions but also all operations development work is included in the priorities. (Pierre SM)

Speed, focus, and efficiency

One answer to the open-ended questions expressed perceived "increased speed" [1] and another mentioned "work better prepared" [1]. One answer claimed that there was "improved efficiency" [1].

Respondents suggested the possibility of having deliveries more frequently [3]. One Agile coach explained:

> [We have] increased the speed of deliveries. Previously it was very rare, like two or three times a year, but now we deliver more often. We had hoped for that, and it is starting to work well. (Alina Coach)

The increased delivery speed was also confirmed by Pierre SM: "We have the opportunity to deliver more often now, they have changed the approval process so that we can release every two months".

Another benefit expressed by respondents was that work was now more focused due to longer plans [4]. Andy SM plainly expressed that: "Much better focus on what work we should conduct and when it must be done". However, one Scrum Master claimed that the improved focus was due to increased freedom once the coaches left Bank:

> It's different now. We are more focused and able to deliver faster than before when the coaches were still here. Then, it was a lot about what you were allowed to do and not allowed to do, but now it's easier to make decisions and focus. (Pierre SM)

Competence dissemination

There were three answers to the open-ended questions which expressed the perceived benefit of "competence dissemination" [3] due to the added inter-team coordination routines. This benefit was not, however, mentioned in interviews or during observations.

Understanding the client perspective

During observations, there were discussions about the presentations at PI planning events. On several occasions, attendees expressed how these gave a better understanding of the client's perspective [5]. During one of the PI planning events, one team member expressed: "It was really good to hear Ester's presentation to understand the customer perspective. It struck me, 'Aha, yes, now I think even more like the customer'".

9.6.2 Perceived drawbacks

The thematic map of codes and themes of perceived drawbacks can be seen in Table 31, organized in the same manner as the perceived benefits above.

Table 31. Codes and themes of perceived drawbacks at Gov.

Codes from open-ended questions	**Codes from observations and interviews**	**Themes**
Too many or too few dependencies [5]	Not enough preplanning [12]	*Not tailored to the work process*
Does not fit operations development work [5]	Implement new routines but keep all old ones [8]	
Not in accordance with the financing procedure [3]	Not enough T-shaped people [7]	
SAFe is not Agile [2]	Cannot plan whole PI because budget not set [7]	
Quality not prioritized [1]		
Too many meetings/overhead [8]	Approval culture [7]	*Too much time spent in meetings*
Inefficient [3]	Too much time in meetings [3]	
Scattered, fragmented, and messy [5]	No one overall responsible [6]	*Unclear responsibilities*
Wrong people in the wrong place [3]	Management lets coaches implement routines [6]	
	Not enough Product Owners [5]	
Only our department works like this [5]	No routines for coordination with other departments [6]	*Clashes with surrounding organizations*
Too little support [1]		
Deadline pressure [2]	Not saving time for unplanned events [7]	*Stress*
	Too many deadlines [5]	
PI plan is not respected [2]	Management bypassing PI planning decisions [9]	*Limited autonomy*
	Management decisions without anchoring with teams [7]	
	Shyness problem with demos, no one dares [4]	*Shyness*

The themes defined for perceived drawbacks are explained below. Numbers in brackets show the number of answers from the open-ended questions that belong to the same code.

Not tailored to the work process

Several informants expressed that there was a problem in using the new routines due to "too many or too few dependencies" [5]. As described above, the definition of dependencies made Gov only highlight a few dependencies, while according to a wider definition, more dependencies

were present. Therefore, based on what the informant viewed as a dependency, some thought there were too many and some to few. Equally, many answers to the open-ended questions expressed that the new way of working "does not fit operations development work" [5]. For the team of operations developers, the added routines were not the only thing new but also the Agile ways of working, which the IT people had used for several years. As described above, the Government Agency could not finish their budgeting process on time for one of the PI planning events. This problem could be seen in three of the answers to the open-ended questions where informants perceived that the routines were "not in accordance with the financing procedure" [3]. Two answers regarding perceived drawbacks from the open-ended questions stated that "SAFe is not Agile" [2]. As described above, although Agile ways of working were new to the operations development team, the other teams at Gov implemented this way of working several years ago. The two answers show that the new routines were perceived as a step backward. One informant also perceived that "quality is not prioritized" [1].

There were many comments during interviews and observations saying that there was not enough preplanning for work that was set during PI planning [12]. One Scrum Master explained the situation:

> We thought the requirements would be better prepared the first time we met for PI planning, but there turned out to be large gaps. Prior to the next PI planning, we realized that we need to work more with the details [of the requirements] for the future than we are used to. (Andy SM)

The quote above was confirmed by two respondents in interviews and in observed discussions during PI planning events. Some quotes from discussions were: "Some teams were unprepared for PI planning", "It is unnecessarily stressful to break down features into stories during PI planning", and "These requirements are just not prepared enough!"

From observations of Scrum of Scrums and PI planning events, there were several heated discussions about the problem of implementing new routines but keeping all the old ones [8]. These two comments from a discussion during a Scrum of Scrums meeting presents the problem:

> Person 1: Should she go to that [project X status] meeting? Why should she spend time on it? I don't understand.
>
> Person 2: We attend all the meetings we did before. Management has not allowed us to stop attending those meetings.

Three respondents said that there were not enough T-shaped people [7], meaning that employees were too narrow in their competence. With too narrow competence, it becomes more difficult to help one another, as Cecilia OpDev explained: "we [operations developers] are too narrow, we don't know enough about technology and IT development. In many meetings, we have been just sitting there looking like fools".

In interviews, three respondents were frustrated regarding not being able to plan for a whole PI since the budget could not be agreed on [7]. One Agile coach explained how they ended up in this situation:

> The budget could not be approved on time. It was at the turn of the year that they should have had everything decided, but with the situation we had last autumn with the government, it was not possible. So that ruined a lot of the planning then, instead of clear long-term direction, we had to wait for directions for a very long time. (Alina Coach)

Although this was a special situation for Gov, a Scrum Master explained how this showed the problem of not tailoring to the current process:

> The budget process does not fit our processes at all. Just because we follow an annual process of budgeting, planning over the turn of the year was completely wrong. We [in our team] only planned for two sprints, for example. Clearly, someone should adjust the thinking for PI planning. (Pierre SM)

Too much time spent in meetings

In the answers to the open-ended questions, the drawback mentioned the most was the perceived drawback of spending time in "Too many meetings/overhead" [8]. Many complained about time wasted and that a lot of the time was only *overhead*. This was perceived as being "inefficient" [3].

Three respondents expressed in interviews that much of these meetings were signs of an approval culture [7], that much effort was spent on seeking approval from managers or Agile coaches. During observations, one example of this approval culture was the number of people attending the *Management Review & Problem-Solving* meeting. The first PI planning event had 35 attendees. Pierre SM explained how this culture turned into too many meetings: "[The problem is] often to get approval from someone. Some manager who calls for a meeting to speak his mind before we can move on."

Spending too much time in meetings [3] was also explicitly expressed in interviews. The quote from Carl Dev is one example: "All these meet-

ings. Meetings, meetings, all the time, long meetings that feel like you are just wasting time".

Unclear responsibilities

Several answers from the open-ended questions described the situation after implementing new routines as "scattered, fragmented, and messy" [5], due to problems of who was doing what. Three informants expressed the problem as there being "wrong people in the wrong place" [3].

Apart from that, respondents in interviews and meetings expressed the problem of no one having the overall responsibility [6]. The following was mentioned during a Scrum of Scrums meeting:

> There are many who take great responsibility for delivering different parts, but few today feel the responsibility for delivering the whole. It is an approach that needs to be adopted in all roles. (Alina Coach)

Also, respondents expressed the problem that management let coaches implement routines [6]. One of the Scrum Masters complained that this was a problem at Gov:

> This is so typical of our organization. The managers leave the responsibility to the coaches to introduce something new so they can either blame them if it goes wrong or take the credit if it goes right. (Andy SM)

The problem is not only expressed by people in the Agile Release Train but is acknowledged by coaches themselves. This quote from Alina Coach is from a meeting with the managers at Gov: "We have pushed on too much work to the coaches. You, managers, must take over the responsibility for the transformation. It cannot continue like this".

In the Scrum of Scrums and interviews, respondents mentioned that Gov did not have enough Product Owners [5]. Carl Dev explained the problem: "We have received so little instructions since we have too few Product Owners. It's been a lot of guesses and assumptions since we don't get a chance to talk to [a Product Owner]".

Clashes with surrounding organizations

Another drawback expressed in the answers to the open-ended questions was that "only our department works like this" [5]. Since this implementation was a pilot project at the Government Agency, other departments had other routines for coordination. Although Gov had several Agile coaches attending during implementation, one answer to the open-ended questions expressed a sense of having "too little support" [1].

Both at meetings and in interviews, respondents expressed that there were no routines for coordination with other departments [6]. The following quote was expressed during a Scrum of Scrums meeting:

> We have two teams [one in another department] that have to decide how to work together because they will work closely together during the next PI. They must have decided this before the PI planning because then we do not have time to sit down and discuss what the collaboration should look like. (Helena RTE)

Stress

In two of the answers to the open-ended questions, respondents mentioned "deadline pressure" [2].

Observed both during PI planning events and in Scrum of Scrums, Gov employees expressed that they were not saving time for unplanned events [7]. As an example, the following discussion took place during one of the Scrum of Scrum meetings:

> Person 1: But is it true that we have maximized all resources in our plan, we have no hours at all for anything else?
>
> Person 2: We were in another situation when we planned before [implementing the inter-team coordination routines], but shouldn't you fill up to a certain percentage and leave space for changes?

In an interview with a Scrum Master, this was confirmed to be an even bigger problem:

> [We had] hopes for how much better we would be at planning, but instead, we have done just as before, we have over-planned and not put in any time to deal with unforeseen events. And when we tried to be realistic and made a plan we believed in, some manager comes in and shifts someone in the team to do something that is said to be more important. (Pierre SM)

Another aspect of the perceived stress expressed in interviews was the occurrence of too many deadlines [5]. Carl Dev explained the situation: "It's been stressful. Too many law changes and deadlines to cope with all the time. It's been tough".

Limited autonomy

Two answers to the open-ended questions regarding autonomy stated that decisions from managers interfered and that the "PI plan is not respected" [2].

This was also put forth in interviews where respondents perceived that management decisions were made without anchoring the decisions in

the teams [7]. An Agile coach expressed this clearly in the following quote:

> Gov has been bad at involving the teams in the decisions. Much has happened over their heads and only been communicated without management listening to the teams. (Alina Coach)

Although not presented in interviews, several employees expressed during observed meetings that management was bypassing PI planning decisions [9]. The following conversation during a Scrum of Scrums meeting is a good example of how this problem was highlighted:

> Person 1: Yes, but when we have agreed on a plan, it is a contract in itself, where you have said, "this is the amount of time we should spend on what the team now plans in their backlog". If you suddenly want to alter that completely during a program increment, that's not something only the manager can decide, as I see it. Because then you can ruin the PI objectives that you have set up in the teams, they can suddenly be ruined.
>
> Person 2: But that is exactly what has happened.
>
> Person 1: Yes, and I say that that's not okay.

Shyness

In observations of Scrum of Scrums, there were discussions on how to conduct system demonstrations. After suggesting that they be conducted with all teams present, one problem addressed was that few people felt comfortable holding presentations in public [4].

Shyness was considered as a problem in how to conduct system demonstrations, since people might not want to present or speak up in front of a larger audience. A Scrum Master describes that this is an actual problem:

> [It has become] more difficult to get people to dare to stand in front of the others and present the plan. With over a hundred people in the room, many bails out because they are too shy. It's actually becoming a problem; people are slinking off when it's time to present the plans. (Pierre SM)

9.7 Changes to team autonomy

In this section, changes to team autonomy as they are perceived by employees at Gov are presented. The perceived changes to team autonomy are analyzed based on whether they support or are in conflict with the Agile principles (Beck et al., 2001).

9.7.1 Choosing features

When it came to freedom of choice of features, several team members addressed their frustration and lack of autonomy. In the *PO synch meetings*, where only Product Owners attended, features were delegated to different teams without involving Scrum Masters or team members. During PI 3, the frustration did not have much to do with being handed certain features, but rather the lack of involvement to understand more of the features before planning them for the PI. The features came as a surprise for the teams since no refinement had been conducted. From this point on, however, more refinement and dividing of features were conducted before the actual PI planning event.

Another example of lack of autonomy from PI planning for PI 3 was the decision on how much the teams were supposed to plan in each sprint. Instead of finding out the need for leeway in the plans for each team, the Agile coaches prescribed to plan for 85 percent of capacity. Even so, Scrum Masters forced the teams to plan for full (100 percent) capacity based on anecdotal evidence of capacity.

These two impacts on team autonomy are not in accordance with AP 5 (*Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done*) and AP 11 (*The best architectures, requirements, and designs emerge from self-organizing teams*). These two Agile principles describe the support and freedom intended for self-organizing teams.

9.7.2 Tradition and culture

At the end of the first PI, Alina Coach wrote a short summary of her observations to the Core Team at Gov. One sentence read: ”One obstacle to continuing towards a higher degree of team autonomy is tradition and culture”. When asked about this sentence, she responded in the following way:

> Whenever someone says that this has been decided, someone will right away ask: “Who has approved that”? It is a culture of finding a person responsible for each decision. (Alina Coach)

Alina Coach further explained that this was not only a problem due to tradition, but also because of rules and regulations within the organization.

> The Agile Release Train is a cross-functional development organization that does not have support in the rules of procedure where mandates are delegat-

> ed to line management. Although the project has not identified any individual obstacle to this, it is important that Gov constitutes the Agile Release Train in line with the rules of procedure and delegates mandate. (Alina Coach)

In other words, the teams in the Agile Release Train at Gov still need a mandate from the line management. The example shows that employees experienced the need for approval without actual mandate in important decisions. This approval culture is not in accordance with AP 5 (*Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done*).

9.7.3 Changed team autonomy along the way

In the beginning, all roles and organizing of teams were decided by people in the Core Team. This changed later when people from the train were invited to suggest changes to the organization. Operations Developers who were originally appointed to different teams were allowed to form a team of their own, thereby giving more autonomy to all teams.

The benefits of involving employees in reorganizing the train were acknowledged by Operations Developers and Alina Coach, who claimed that this made them move towards more autonomy in all teams. Alina Coach further expressed that proposed changes during PI 3 were not met with resistance in the same manner as earlier, due to employee involvement.

During the last PI, teams perceived more autonomy due to the Agile coaches leaving the train. Several comments during the PIs from team members as well as the Core Team expressed that they thought the Agile coaches implemented SAFe routines by the book, without trying to adapt to the environment at Gov. Especially, the program board artifact was seen as superfluous since the number of dependencies was so few within the train. When the Agile coaches were not allowed to remain, people at Gov experienced that they suddenly had more freedom to act on their own and make decisions on how to work in their teams without asking coaches for approval.

In answering the open-ended survey question about the major drawbacks of this new way of working, two people mentioned limited autonomy. Although that does not give a full picture of perceived autonomy at Gov, it is significant when *limited autonomy* is expressed as the *main drawback* of the new way of working.

10 Cross-case analysis

In the three previous sections, I have analyzed single cases of inter-team coordination routines in large-scale ASD settings. Context has been key to describe and explain tailoring and impacts for every single case.

Studies sometimes focus too much on each case and try to answer the research questions through single cases, instead of investigating the phenomenon by searching for patterns between multiple cases (Yin, 2017). The aim of this section is, therefore, to investigate and analyze all three cases by conducting what Yin calls a *cross-case synthesis* (ibid.).

By selecting, partitioning, and clustering data from the three cases, I will present several matrices so that contrasts between cases become clearer. This procedure sometimes requires a further transformation of the data from the three cases into short quotes, summarizing phrases, ratings, and symbols (Miles et al., 2014). These matrices help in visualizing the data of the individual cases but also in presenting similarities and differences between them.

The first section in this chapter presents differences and similarities in the characteristics of the case organizations. The second section (10.2) presents institutional logics in order to answer underlying reasons for differences in tailoring, thereby to some extent address RQ2 (*Why are inter-team coordination routines tailored (or not tailored)?*). The third section (10.3) presents the dependency types in the cases based on coordination theory.

This section also addresses RQ2 to some extent in providing reasons for inter-team coordination. In the fourth section (10.4), similarities and differences in implementation and tailoring of inter-team coordination routines are presented to address both RQ1 (*How are inter-team coordination routines performed and tailored in large-scale Agile software development?*) and RQ2.

Sections 10.5 and 10.6 present the perceived impacts of the implemented routines in all the cases to answer RQ3 (*What perceived impacts (benefits and drawbacks) are associated with the implementation of inter-team coordination routines?*). The last section (10.7) displays differences and similarities in perceived changes to team autonomy to answer RQ4 (*How is team autonomy perceived to be changed after implementing inter-team coordination routines?*).

10.1 Characteristics of the case organizations

To understand the background and history of the studied cases, a comparison of characteristics is warranted. Therefore, Table 32 presents a clustered summary of the characteristics of the case organizations.

Table 32. Case-level matrix of characteristics of the case organizations.

Characteristics	**Auto**	**Bank**	**Gov**
Industry	Automotive	Banking	Government agency
Started implementing ASD	2012	2013	2014
Started implementing inter-team coordination routines	January 2017	April 2017	March 2017
Agile Release Trains	3	1	1 later split into 2
Number of teams	From 24 to 31	From 5 to 8	From 5 to 8 until the split into two trains where train 1 consisted of 3 teams and train 2 of 5 teams
Reasons for implementing inter-team coordination routines	Improvements in planning and coordination on a team-of-teams level. Increased personal, overall responsibility for the product	Improvements in planning and coordination on a team-of-teams level.	Consensus on goals and direction. Better overview and increased transparency. Even workload between teams.

10.1.1 Similarities in all three cases

It is no surprise that the contextual components of each case are diverse since a maximum variation approach was chosen. However, something that all cases do have in common is the timing of the introduction of both Agile ways of working and inter-team coordination routines. They all introduced Agile ways of working sometime between 2012 and 2014, and they all started implementing inter-team coordination routines during the first part of 2017. Also, the average number of years employees had been at the organizations was similar, ranging from 8.4 to 9.6 years.

10.1.2 Similarities in two of the cases

Regarding the reasons for implementing inter-team coordination routines, both Auto and Bank expressed a need to improve planning and coordina-

tion on a team-of-teams level. This goal is not very surprising. Gov on the other hand, did not state these improvements as a reason, but instead expressed a need for consensus on goals and direction, as well as transparency and overview.

10.2 Comparing institutional logics

In the previous chapters, signs of institutional logics have been presented in each of the three case organizations. These logics have been exemplified with quotes and notes from observations depicting the view in the studied departments. Table 33 presents a case-level matrix of the identified institutional logics.

Table 33. Case-level matrix of institutional logics.

Institution	**Auto**	**Bank**	**Gov**
Agile ways of working	Agile toolbox logic	Agile toolbox logic	Agile rulebook logic
Product development	Flow efficiency logic	Hybrid (Resource/Flow) efficiency logic	Resource efficiency logic
Response to survey statements (Likert scale 1-5: 1=Not true, 5=Very true)	**Auto**	**Bank**	**Gov**
"There are many problems with working in a large-scale setting as ours, using SAFe"	Mean: 3.3 SD: 1.0 Median: 3 Mode: 3	Mean: 3.2 SD: 0.9 Median: 3 Mode: 4	Mean: 3.9 SD: 0.8 Median: 4 Mode: 4
"I would like to go back to the old way of working (instead of working according to SAFe)"	Mean: 1.9 SD: 1.1 Median: 2 Mode: 1	Mean: 1.9 SD: 1.1 Median: 2 Mode: 1	Mean: 2.6 SD: 1.2 Median: 3 Mode: 4

10.2.1 Similarities in two of the cases

The three cases did not share the same institutional logics. That is not surprising since one of the criteria in selecting cases was to find organizations with diverse business logics. One might, therefore, expect that different business logics could mean differences in institutional logics as well. There are, however, some similarities.

Investigating the community logics of Agile ways of working, the study showed that Auto and Bank shared an *Agile toolbox logic*. Routines were chosen, tailored, and sometimes new routines were invented to suit the situation. The two cases also shared an overall view on implementing the new routines based on the survey responses to the two questions

(*There are many problems with working in a large-scale setting as ours, using SAFe* and *I would like to go back to the old way of working (instead of working according to SAFe*)). They both acknowledged a somewhat problematic situation in implementing the routines (an average of 3.3 at Auto and 3.2 at Bank), but few employees would like to go back to the previous way of working without the implemented routines (an average of 1.9 and Mode of 1 at both Auto and Bank).

Regarding the corporate logic of product development, Gov and Bank, to some extent, share a *resource efficiency logic* that contradicts the values of SAFe. Although, as the institutional logic perspective presumes, categorizing within an ideal type does not mean that a case is always anchored in that ideal type (Thornton et al., 2012). Dominating logics in a case is fluid and changes over time. This was especially evident at Bank, and to some degree at Gov, who both started to show signs of moving towards flow efficiency logic at the end of the studied period. Bank moved so much towards flow efficiency logic that they could rather be categorized as a hybrid logic, with two coexisting logics during the observed time period. That is not surprising since the values of SAFe supports a flow efficiency logic, and routines are prescribed to foster a flow efficiency mindset. For example, the PI is prescribed to contain a final Innovation and Planning sprint, allowing time for improvements and innovation (Scaled Agile, 2020). Increasingly more time for Innovation and Planning was allowed at both Bank and Gov.

10.2.2 Differences between the cases

Gov was the only case to be dominated by *Agile rulebook logic*, where coaches forced the organization to implement routines by the book. Analyzing employee interview answers and comments during observations, this evidently caused frustration among the teams. It was not until the coaches left the organization that routines became tailored, and sometimes dropped, and the teams started to move closer to *Agile toolbox logic*. Based on the survey questions, employees at Gov showed the most negative view towards the implemented routines. More employees experienced difficulties in implementing the routines in their environment compared to Auto and Bank (an average of 3.9 to the statement: *There are many problems with working in a large-scale setting as ours, using SAFe*). Also, more employees at Gov wanted to return to the former way of working, compared to Auto and Bank (an average of 2.6 and a Mode of 4 to the statement *I*

would like to go back to the old way of working (instead of working according to SAFe)).

Based on institutional logics, this suggests that a combination of *Agile rulebook logic* and *resource efficiency logic* might be problematic when implementing inter-team coordination routines in large-scale ASD settings.

10.3 Types of dependencies

A suggested starting point for analyzing coordination, according to coordination theory (Malone & Crowston, 1994), is to investigate what kind of dependencies the organization is facing. The different sorts of dependencies have been presented for each different case previously in this thesis. Below, Table 34 shows a case-level matrix of dependency types.

Table 34. Case-level matrix of dependency types.

Dependency types	**Auto**	**Bank**	**Gov**
Types according to coordination theory	Producer-consumer, subtypes transfer and precedence	Producer-consumer, subtypes transfer and precedence	Producer-consumer, subtypes transfer and precedence
View of sequential or pooled dependencies	From all sequential to mostly pooled	From mostly sequential to mostly pooled	Mostly pooled
Static versus emerging dependency issues	From almost none to estimated 25 percent emerging	From an estimated 10 percent to 50 percent emerging.	Calculated 23.5 percent emerging *sequential* dependencies.

From a coordination theory perspective (Malone & Crowston, 1994), all three cases dealt with the same types of dependencies. The dependencies were of *producer-consumer* type, were the subtypes *transfer* and *precedence* were identified. This means that 1) all three organizations encountered dependencies where one team could not start until another had finished their part (*transfer*), and 2) had a need to recognize in advance when a job was possible to continue (*precedence*). This suggests that these kinds of dependencies could be identified in other large-scale ASD organizations.

In analyzing dependency types, according to Thompson's (1967) classification, *sequential* and *pooled dependencies* are recognized. The third type, *reciprocal dependency*, is not. When investigating the amount of sequential versus pooled dependencies (Thompsons, 1967), a difference can be identified both between cases as well as within each case. In both case Auto and Bank, respondents presented different views of most dependencies.

Some respondents experienced most dependencies to be sequential. At Auto, some even claimed that all dependencies were sequential, while others perceived most dependencies to be pooled. This differs from case Gov where the majority of dependencies could be identified as pooled dependencies. Albeit the difference, the commonality of all three cases is that both sequential and pooled dependencies were present and important to manage.

Regarding whether dependencies were easy to identify in advance or whether emerging dependencies were common, there were differences between the cases as well. At Auto, the perceived amount of emerging dependencies ranged from almost none to an estimated 25 percent. At Bank, the corresponding number was an estimated 10 to 50 percent, which was the highest reported amount. At Gov, 23.5 percent *sequential* dependencies could be calculated as emerging dependencies. However, this does not show the full picture at Gov, since pooled dependencies are not accounted for. The narrow definition of dependencies at Gov only allowed sequential dependencies to be accounted for as real dependencies.

The commonality between the three cases is that emerging dependency issues were common and important to manage. Okhuysen and Bechky (2009) claimed that too much of previous research has focused on static ways to manage coordination, where dependency issues can be identified in advance and managed through good planning. They argue that more attention to solving emerging dependency issues is called for. By investigating three different implementations, this data strengthens their argument in the domains of large-scale ASD. The same pattern could be seen in three organizations with diverse institutional logics.

10.4 Inter-team coordination routines

This section is divided into several subsections where each subsection presents an analysis of three inter-team coordination routines; PI planning; the Scrum of Scrums routine; and routines for using and managing the program board. For each routine, a comparative overview between the three cases is presented in matrices, together with descriptions of how SAFe prescribes the performance of each routine. Apart from presenting performance and tailoring of the routines, the matrices also show results from the survey question on each routine.

10.4.1 The PI planning routine

The PI planning routine, as described by SAFe (Scaled Agile, 2020), is presented in a static way. The PI period to plan for is supposed to have a set cadence based on a fixed number of calendar weeks, and the suggested standard agenda is described in detail. According to SAFe, the most common cadence is to plan for four two-week sprints, followed by a two-week Innovation and Planning sprint. Nothing is mentioned regarding developing or tailoring of the routine itself, only the prescribed format. This prescribed format was not followed to the letter in any of the three organizations studied. Below, Table 35 presents a case-level matrix for the PI planning routine.

Table 35. Case-level matrix for the PI planning routine.

Routine	**According to SAFe**	**Auto**	**Bank**	**Gov**
PI planning	Cadence: Set cadence, PI plan every 8-12 weeks.	Cadence: Set cadence, PI plan every 10 weeks.	Cadence: Started with 5 weeks, finally set at 9 weeks cadence.	Cadence: 12 weeks for most. 15 weeks for PIs stretching over holidays
	PI duration: 4*2-week sprints and IP sprint (2 weeks).	PI duration: 3*3-week sprints and IP sprint (1 week).	PI duration: First 2*12-day sprints, finally set at 4*12 and 2 days IP sprint.	PI duration: 4 sprints of 3 weeks length, no IP sprint. 5 sprints for PIs stretching over holidays
	Planning event duration: 2 days.	Planning event duration: 1.5 days.	Planning event duration: From 0.5 to 1.5 days.	Planning event duration: 2 days.
	Team breakout time: 31 % (5h/16h)	Team breakout time: From 45.8 % to 65.2 % (final obs)	Team breakout time: From 52.5 % to 62.5 % (final obs)	Team breakout time: From 30 % to 57.1 % (final obs)
Survey Question (Likert scale 1-5: 1=Not true, 5=Very true)	*"The PI planning gives me a good overview of our work"*	Mean: 4.2 SD: 0.7 Median: 4 Mode: 4	Mean: 3.9 SD: 0.5 Median: 4 Mode: 4	Mean: 3.2 SD: 1.0 Median: 3 Mode: 4

Similarities in all three cases

There were some similarities in all three cases. One thing was that the number of teams within each train changed between each presented act,

and sometimes between each PI planning. In that sense, the Agile Release Train could be called a temporary organization (Lundin & Söderholm, 1995). In their theory of temporary organizations, Lundin and Söderholm (ibid.) mainly refer to projects as temporary organizations. Projects are referred to as a way to organize people for a one-time, specific purpose, where one of the challenges is teamwork in newly formed teams. The Agile Release Train seems to be a similar bird, yet not quite of the same species. Teams are protected in trying to have a stable environment with the same team members for as long as possible.

The temporality is not the team itself but rather the composition of teams between each PI, the Agile Release Train. Hence, the Agile Release Train is an example of a temporary organization. Although there is no outspoken lifetime for the train as such, each PI has a defined end where the train can be stopped or split into two, which happened at Gov. This means that the only real defined lifespan for an Agile Release Train is one PI at a time.

Another similarity in all cases was the increased amount of team breakout time in relation to overall planning time. The standard agenda prescribed by SAFe contains a number of items focusing on presentations and reviews, and only 31 percent is suggested for team breakout time. Although Gov started in the same manner as suggested by SAFe, none of the cases ended up spending as much time in presentations and reviews. Rather, they all tailored in favor of utilizing time for team breakouts.

Even though SAFe suggests a full sprint at the end of each PI in order to allow time for both *Innovation and Planning*, none of the organizations were allowed a whole sprint for this. Auto was allowed one week while both Bank and Gov were not allowed more than a short *Inspect and Adapt* meeting apart from the PI planning workshop for the upcoming PI.

One similar impact that did not have to do with implementation or tailoring was that planning precision increased in all three cases. This was not as easy to spot at Gov with the extensive reorganization happening, but since the train was divided into two, increased planning precision was easily detected at both Auto and Bank.

Similarities in two of the cases

Some tailoring of the PI planning routine could be seen in two out of the three cases. SAFe prescribes all teams to attend physically during PI planning, but both Auto and Gov had remote teams. With teams in different cities and even countries, they both decided it was not financially feasible

to transport teams to one location for every single PI planning event. Although that caused some difficulties, it was not particularly problematic for conducting the PI planning routine. Looking at the perceived drawbacks (chapter 11.3.2), none is related to the distributed setup neither at Auto nor at Gov.

Cases Auto and Bank did not need two full days for PI planning. One-and-a-half day was sufficient, even for periods as long as ten weeks. SAFe suggests the PI length to last between eight and twelve weeks and still prescribes two days for planning.

Both Bank and Gov implemented a "coordination ownership" for features as a way to tailor the coordination routine. Bank named this convention *full feature ownership*. That meant that instead of only finding out and solving dependency issues between teams, team ownership of a feature was implemented. Thereby, one team was given the responsibility to follow up on dependencies for one specific feature, without raising dependency concerns with the whole Agile Release Train during the PI. This tailoring decision was implemented in two of the cases without any knowledge of one another.

Differences between the cases

One thing that differed between all three cases was the logic behind the length of the planning period. At Auto, a set number of calendar weeks was chosen in the same manner as SAFe prescribes (Scaled Agile, 2020). Bank chose a flexible approach to decide on the planning period by starting small, only planning for two sprints. They thereafter increased the planning period and finally settled on four sprints. The logic behind gradually increasing the planning period was to learn the PI planning routine faster, since they occurred close to each other for the first number of PIs. Finally, at Gov, instead of focusing on calendar weeks, the number of sprints to plan for was based on available man-hours. This meant that Gov decided on to allow for more or fewer weeks depending on interfering holidays.

Another difference between the cases was how much they used Agile coaches in implementing and tailoring their routines. Auto did not use any Agile coaches except for during initial discussions before starting the implementation. Bank had one Agile coach for the full observed time period. He initially took the role of Release Train Engineer during the first PI:s but later let an employee fill this role. Instead, he remained as an Agile coach for the remainder of the time. Gov used the most amount with

five Agile coaches and an additional two coaches working with leadership and team development. Even so, many complained about not getting enough help from the coaches at Gov. In this case, routines were very much implemented according to SAFe, adhering to an *Agile rulebook logic*. I argue that this might be the case when several coaches attend. Much of the discussions between coaches at Gov related to whether something was *correct* according to SAFe or not. For as long as the Agile coaches remained, the framework became the norm, and team members avoided attempts to invent and adapt routines on their own.

Regarding the survey question (*The PI planning gives me a good overview of our work*), it shows that Auto viewed this routine as most beneficial (Mean = 4.2), followed by Bank (Mean = 3.9) and finally Gov (Mean = 3.2). Although the question only related to one aspect of PI planning (*overview*), the result suggests that Auto and Bank, who conducted the most amount of tailoring to the routine, were most satisfied in using the routine.

10.4.2 Scrum of Scrums

SAFe describes the Scrum of Scrums routine in a static way with a fixed timespan and suggests a standard meeting agenda (Scaled Agile, 2020). Although the agenda is presented as an example, nothing regarding developing or tailoring of the routine is mentioned. Below, Table 36 presents a case-level matrix for the Scrum of Scrums routine.

Table 36. Case-level matrix for the Scrum of Scrums routine.

Routine	**According to SAFe**	**Auto**	**Bank**	**Gov**
Scrum of Scrums	Cadence: Twice weekly.	Cadence: Varied from once per sprint to once a week.	Cadence: Daily + mid-sprint review.	Cadence: Once a week.
	Duration: 30 minutes and "meet after".	Duration: 20-30 min, no "meet after".	Duration: 15 min, sometimes "meet after".	Duration: 30 min, no "meet after".
	Attendees: RTE, SMs, and possibly stakeholders.	Attendees: RTE and SMs.	Attendees: RTE, SMs, some stakeholders, and some team members.	Attendees: Managers, RTE, and SMs.
Survey Question (Likert scale 1-5: 1=Not true, 5=Very true)	*"The Scrum of Scrums meetings solve inter-team coordination problems"*	Mean: 3.3 SD: 1.1 Median: 4 Mode: 4	Mean: 3.4 SD: 0.8 Median: 4 Mode: 4	Mean: 3.2 SD: 0.8 Median: 3 Mode: 4

Similarities in all three cases

What was similar between the three cases was that they did perform something they chose to call Scrum of Scrums. Another similarity was that no one decreased the frequency of meetings along the way. Beyond that, nothing was similar. Even so, none of them performed Scrum of Scrums as they are prescribed by SAFe in their standard agenda (Scaled Agile, 2020). The survey question (*The Scrum of Scrums meetings solve inter-team coordination problems*), shows that Auto (Mean = 3.3), Bank (Mean = 3.4), and Gov (Mean = 3.2) were quite similar regarding their view on the usefulness of the Scrum of Scrums routine.

Similarities in two of the cases

Something that Auto and Bank had in common was the increased frequency of Scrum of Scrums along the way. Both Auto and Bank reported that they realized the increased importance of this meeting, thereby increasing the frequency. During the final PIs at Auto, they settled for a cadence of one Scrum of Scrums per week, which was the same cadence as at Gov.

Differences between the cases

What differed at Gov compared to the other cases, but made the routine similar to what SAFe prescribes (ibid.), was that they did not perform any tailoring along the way. However, the way the routine was set up was not according to what SAFe prescribes. Gov performed their Scrum of Scrums more like a traditional status meeting, spending much more time on information-sharing than solving dependency issues.

One could even question whether the meeting they performed should be called Scrum of Scrums or whether this was a typical example of an organization falling back into old habits. The routine of having weekly meetings with information and status updates was the way projects had previously been handled at Gov.

10.4.3 Program board

According to SAFe, the program board should be used to highlight feature delivery dates, and feature dependencies among teams, and show relevant milestones (ibid.). The program board is presented by SAFe as a physical board containing paper post-it notes and red-colored strings showing feature dependencies between teams. The board may, or may not, be maintained manually according to SAFe (ibid.). Nothing is mentioned regarding developing or tailoring the use of the program board.

Below, Table 37 presents a case-level matrix for using and managing the program board.

Table 37. Case-level matrix for using and managing the program board.

Routine	According to SAFe	Auto	Bank	Gov
Program board	Contains: Feature delivery dates, feature dependencies, and milestones.	Contains: Feature dependencies and milestones, not all features.	Contains: Feature delivery dates, feature dependencies, and milestones.	Contains: Feature delivery dates, feature dependencies, and milestones.
	Up to date: May or may not be maintained during PI.	Up to date: Not maintained during PI.	Up to date: Updated every sprint planning day and during mid-sprint review.	Up to date: Not maintained during PI. Later abandoned.
	Format: A physical board	Format: A digital board	Format: A physical board	Format: First physical, later digital, later abandoned.
Survey Question (Likert scale 1-5: 1=Not true, 5=Very true)	*"The program board is very helpful for coordination between teams"*	Mean: 3.4 SD: 1.0 Median: 3 Mode: 3	Mean: 3.3 SD: 0.9 Median: 3 Mode: 4	Mean: 2.8 SD: 0.8 Median: 3 Mode: 3

Similarities in all three cases

What was similar between the three cases was that they all started to use a program board to visualize dependencies. Apart from that, nothing was similar in all three cases.

Similarities in two of the cases

One thing that Auto and Gov had in common was that none of them kept the program board up to date during the PI. The routine to visualize dependencies was performed during PI planning, but no routine was performed to keep the program board up to date after that.

Differences between the cases

Something that stood out at Bank was the outspoken increased importance of the program board. The Release Train Engineer referred to it as the "master", where all planning should be visualized. This can be compared to Gov, the only one of the three cases where the use of the program board was entirely abandoned. This decision was made since few

dependencies were visualized on the program board; hence they did not see the added value of visualizing dependencies. That could to some extent explain the results from the survey question (*The program board is very helpful for coordination between teams*), which showed that Gov (Mean = 2.8) viewed the program board as less helpful, while Bank (Mean = 3.3), and Auto (Mean = 3.4) were quite similar regarding their view on the usefulness of the program board. For Bank, the use of the program board led to an invention of a new routine to keep it up to date: the mid-sprint review. SAFe does not give any guidance on how to keep the program board up to date, only that it is optional for the organization to do so (Scaled Agile, 2020). Bank did not think that this guideline was helpful enough. At Bank, this clearly showed the importance of managing emerging dependency issues, as described by Okhuysen and Bechky (2009).

10.5 Perceived benefits

In the previous three chapters, themes of perceived benefits have been identified and discussed in each case. In this section, the perceived benefits are discussed in relation to my systematic literature review of other published studies, presented in section 2.5. Table 38 presents a case-level matrix of perceived benefits.

Table 38. Case-level matrix of perceived benefits.

Perceived benefits	**Auto**	**Bank**	**Gov**
Similarities in all three cases	Overview and transparency	Overview and transparency	Overview and transparency
	Coordination and cooperation	Coordination and cooperation	Coordination and cooperation
	Speed, focus, and efficiency	Speed, focus, and efficiency	Speed, focus, and efficiency
	Auto	**Bank**	**Gov**
Similarities in two of the cases	Planning precision	Planning precision	
	Employee motivation	Employee motivation	
	Clear priorities		Clear priorities
	Auto	**Bank**	**Gov**
Unique for one case	Stress relief	Less dependent on individuals	Competence dissemination
	No interference on teamwork		Understanding the client perspective

10.5.1 Similarities in all three cases

The two most reported benefits in the open-ended questions of the survey in all three cases were 1) the perceived improvement of overview and transparency and 2) improved coordination and cooperation. These benefits were also described in interviews. In the conducted systematic literature review, *overview and transparency* was the second most reported benefit in other studies of large-scale ASD implementations (see Table 2). At Gov, the improved overview and transparency were explicitly stated as a reason for implementing inter-team coordination routines, which shows this to be an expected benefit.

Coordination and cooperation were the most reported benefits in other studies (see Table 2). It is not surprising to find this perceived benefit common to all three studied cases since, after all, coordination is the intended purpose of inter-team coordination routines.

The systematic literature review presented *speed, focus*, *and efficiency* as the third most often reported benefit in the identified studies, identified in five different papers (see Table 2). In the three cases, no more than eight of the open-ended answers related to speed. However, speed was more frequently mentioned in interviews and meetings. Also, speed was not only related to work itself, i.e. developing software, but also to the increased speed of planning compared to the previous way of working. This showed the perceived benefit of the PI planning routine.

To summarize, these three benefits identified in all three cases were also identified in previous studies on benefits in large-scale ASD. From looking both at the open-ended answers as well as the number of codes from observations and interviews relating to perceived benefits, a clear pattern is recognized. The two most reported benefits are the ones that are common to all three cases. This suggests that these perceived benefits, 1) overview and transparency, and 2) coordination and cooperation, might be expected in implementations of inter-team coordination routines, regardless of context.

10.5.2 Similarities in two of the cases: Auto and Bank

Auto and Bank shared two perceived benefits: 1) increased planning precision, and 2) improved employee motivation. As presented in the narrative describing implementation at Bank, planning precision was identified in observations for each PI. Finished work versus planned work increased from 60 percent in PI 0 to 87 percent in PI 5. At Auto, the benefit of

planning precision was expressed in eleven of the open-ended answers in addition to what the numbers showed. The numbers showed that finished work versus planned work increased from 78.4 percent in PI 3 to 100 percent in PI 10.

Is there a reason why Gov did not perceive the benefit of increased planning precision? I argue that there are three reasons: 1) the extensive reorganization that took place, 2) difficulties in measuring progress, and 3) the institutional logic in play. The first reason is that they divided the department into two Agile Release Trains early in the implementation process. This change made the cooperating teams, now in new constellations, to have to learn about their total planning capacity all over again. This, to some degree, also gave rise to the second problem of difficulties in measuring progress, since the newly formed trains had no history of measurements. But even so, Gov also changed how to measure progress along the way even before splitting the train, which might have caused difficulties in perceiving an increase in planning precision. Regarding the third reason, the *resource efficiency logic* at Gov made teams plan for full capacity, leaving no leeway in their plans. As presented in the narrative at Gov, this happened even when asked by the coaches to plan for only 85 percent of full capacity. With unrealistic plans, Gov did not learn from previous mistakes, and the measured planning precision did not increase.

Employee motivation was mentioned in different ways at Auto and Bank. Apart from what was highlighted in interviews, five open-ended answers at Auto explicitly expressed perceived increased motivation. At Bank, the increased motivation was expressed in joyous terms in the open-ended answers, such as "g*reat fun!"* Increased employee motivation was one of the identified benefits reported in six of the identified previous studies on large-scale ASD (see Table 2). This study, therefore, further confirms employee motivation to be a possible outcome.

10.5.3 Similarities in two of the cases: Auto and Gov

Auto and Gov also shared the perceived benefit *of clear priorities*. The benefit *clear priorities* was not identified in any of the studied papers in the SLR. Perhaps a reason for this lack of identified benefit is due to how things are presented. One could argue that increased speed, for example, could be an effect of clear priorities. In such cases, clear priorities might not have been mentioned but instead seen as a prerequisite for the increased speed.

No matter the reason, the identified perceived benefit of *clear priorities* is an important finding of this study worth highlighting. Despite identified drawbacks of spending much time in meetings (more on that in the next section on perceived drawbacks), many perceived that priorities were clearer.

One aspect of prioritization was mentioned at Auto. Despite of making longer plans for one PI at a time, the method of working in sprints afforded flexibility to change priorities when needed. Even if it seemed like plans were committed for a longer timeframe than before, more opportunities to change priorities arose in each new sprint and planned work could be put back in the product backlog when needed.

However, this benefit was not perceived at Bank. Could it be due to differences in institutional logics? I argue that this is not so. The answer might be found in an event presented in the narrative about PI planning in PI 3 at Bank. During this PI planning event, a misunderstanding of priorities led to wasted time and plans. This, in turn, would not have happened if the program board had been up to date. Because of this event, employees at Bank did not perceive clear priorities as a benefit at this stage – but rather the contrary.

10.5.4 Differences compared to previous studies

Figure 19 shows a mapping of perceived benefits identified in this study versus findings in previous studies. The green-colored boxes mean that the benefit is identified in previous studies as well as in this study. The order of displayed benefits is based on frequency, i.e. the most reported benefits are at the top.

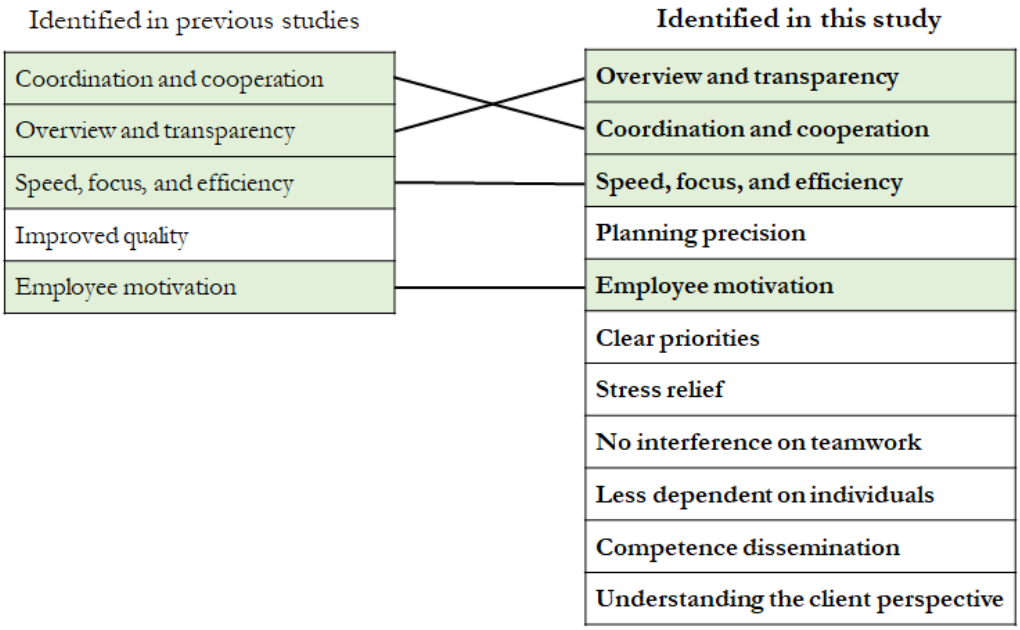

Figure 19. Benefits identified in previous studies vs. this study (ordered by frequency).

One benefit put forth in five of the previous studies (see Table 2) but not in this study is *improved quality*. This is somewhat surprising, and one might speculate why that has not been brought up in any of the cases studied. I argue that the concept of *quality* might be inherent in reported benefits, such as increased *speed, focus, and efficiency*, maybe even in benefits such as *planning precision*.

There are several perceived benefits identified in this study that are not presented in previous studies (as presented above in Figure 19). The two impacts of *stress relief* and *no interference on teamwork* paints a picture of a better working climate, despite the ongoing transformation in the organization. These benefits were reported at Auto, especially by managers and Release Train Engineers, but also by many team members.

Less dependent on individuals was only identified at Bank, and *competence dissemination*, and *understanding the client perspective* was only identified at Gov. These single identified benefits in only one case might be context specific. Further research is needed to investigate whether that is the case.

10.6 Perceived drawbacks

Table 39 presents a case-level matrix of perceived drawbacks in the three investigated cases.

Table 39. Case-level matrix of perceived drawbacks.

Perceived drawbacks	**Auto**	**Bank**	**Gov**
Similarities in all three cases	Too much time spent in meetings	Too much time spent in meetings	Too much time spent in meetings
	Not tailored to the work process	Not tailored to the work process	Not tailored to the work process
	Unclear responsibilities	Unclear responsibilities	Unclear responsibilities
	Clashes with surrounding organizations	Clashes with surrounding organizations	Clashes with surrounding organizations
	Auto	**Bank**	**Gov**
Similarities in two of the cases	Stress		Stress
	Shyness		Shyness
	Less ad-hoc coordination	Less ad-hoc coordination	
		Limited autonomy	Limited autonomy
	Auto	**Bank**	
Unique for one case	Lacking sense of overall responsibility	Not recognizing individual contributions	
	Long-term planning difficulties	Risk of exclusion	

10.6.1 Similarities in all three cases

The drawback mostly mentioned in the open-ended answers was frustration regarding *too much time spent in meetings*. Especially, many complained about having to attend too many meetings. Some perceived that spending so much time in meetings made the progress of work too slow. This drawback was also identified in two previous studies in the SLR (see Table 2). Also, Stray (2018) presented in a case study at a large-scale ASD company that the average time spent on scheduled and unscheduled meetings for team members was as much as 2.7 hours per day.

The second most mentioned perceived drawback in the open-ended answers to the survey was that the implemented routines were *not tailored to the work process*. Several voices said that they perceived that too much was implemented by the book without tailoring to the specific conditions of their organization. This drawback correlates to *routines not helpful for teams*, found in eight previous studies identified in the SLR (see Table 2). However, the theme in this study, *not tailored to the work process*, is wider and includes more components apart from the routines that were not helpful

for teams, such as routines not helpful for other roles, e.g. Product Owners, Release Train Engineers, and Epic Owners. Several answers to the open-ended question regarding drawbacks due to the added routines were a variation of *this is not Agile*. For example, several employees at Gov viewed the implementation of routines such as Scrum of Scrums as a step back to old habits of typical status meetings. This confirms the importance of tailoring of routines and the risk of implementing by the book, adhering to an *Agile rulebook logic*. Also, it presents a possible expected drawback of implementing inter-team coordination routines.

The perceived drawback *unclear responsibilities* was, in many cases, described as a clash between new and old roles. Studies show (Bass, 2015; Gustavsson, 2017) that there is a vast number of roles involved in different setups of large-scale Agile settings. The same role has even had different authorities and responsibilities in different organizations (Gustavsson, 2017). New and unclear roles might result in misunderstandings (Stray et al., 2018). Also, something to highlight from the Gov case was the clash between Agile coaches versus managers in the organization where several voices raised the concern of managers leaving implementation in the hands of coaches. This drawback confirms previous research since the same thing was identified in four studies in the SLR (see Table 2).

From looking both at the open-ended answers as well as the number of codes from observations and interviews relating to perceived drawbacks, the same pattern as with perceived benefits is recognized. These three presented perceived drawbacks, identified in all three cases, were the ones most often reported. This suggests that these three most common perceived drawbacks might be expected in implementations of inter-team coordination routines, regardless of context.

The fourth identified perceived drawback appearing in all three cases was that the implemented inter-team coordination routines sometimes *clashes with the surrounding organizations*. This drawback shows the absence of prescribed routines in SAFe to manage dependencies outside the Agile Release Train. This drawback was identified in two papers in the SLR (see Table 2) and further confirms previous research.

10.6.2 Similarities in two of the cases

One drawback highlighted in seven of the previous studies investigated in the SLR was increased level of *stress* (see Table 2). This perceived drawback was expressed in open-ended answers but only in four survey answers at Auto (out of 125 survey answers). However, it is noteworthy to

discuss how different things are perceived by people in the same case organization. Although a few respondents reported experiences of stress, employees at Auto also mentioned *stress relief* as a perceived benefit. This was especially prevalent in interviews with managers who iterated voices of thankful employees. Some employees explicitly expressed the implementation of inter-team coordination routines to be the reason for a less stressful environment. One conclusion from Auto might be that although the implementation caused less stress for most employees, some still had a stressful work situation. One possible reason for experiencing stress is suggested by Meier et al. (2018) who identified that employees new to Agile experienced higher levels of stress.

At Gov, on the other hand, several people reported stress, especially due to added work by following the prescribed details of SAFe forced on them by Agile coaches adhering to an *Agile rulebook logic*. They also reported stress due to not allowing time for leeway in plans and redundancy due to the *resource efficiency logic* at Gov. This suggests that a combination of *Agile rulebook logic* together with *resource efficiency logic* might increase stress when implementing inter-team coordination routines.

Another perceived drawback at both Auto and Gov was *shyness*. One might think that shyness would not be a real problem in a software development environment, but there are practical issues related to it. For example, parts of the PI planning routine call for public presentations of your plans. Compared to before the implementation of this routine, team members only needed to present plans within their own team. In both presentations of the Final plan in PI planning as well as in Scrum of Scrums, employees are expected to speak up about problems. If individuals are too shy to speak up, this could cause real problems. This drawback was also identified in two previous studies (see Table 2). This study further confirms the possible drawback of shyness in large-scale ASD settings.

One drawback identified at both Auto and Bank was *less ad-hoc coordination*. In other words, apart from the implemented routines for inter-team coordination, there was limited interaction between teams. However, as discussed in Stray (2018), although the team members in the case study spent an average of 1.7 hours per day in un-scheduled meetings, they often underestimated the time they spent in meetings, which is common (McCall, Morrison, & Hannan, 1978). I argue that this perceived drawback relates to the number of open-ended answers expressing that too much time was spent in meetings. With too much time spent in meet-

ings, there might be a resistance to conducting ad-hoc meetings as well. This was not identified in any of the previous studies and is an important finding in this study.

At Bank and Gov, one drawback was also identified in both cases. Some people perceived *limited autonomy*. One open-ended answer at Bank expressed that teams felt micro-managed, and at Gov, open-ended answers expressed that sprint plans where not respected. Interview answers showed that managers bypassed PI planning decisions and micro-managed team members without anchoring decisions in the whole team. Limited autonomy was also identified as a perceived drawback in five previous studies, as presented in the SLR (see Table 2). The changes to autonomy will be further discussed in a section below (11.3.3).

10.6.3 Differences compared to previous studies

Below, Figure 20 presents a mapping of perceived drawbacks identified in this study versus findings in previous studies. The red-colored boxes mean that the drawback is identified in previous studies as well as in this study. The order of displayed drawbacks is based on frequency, i.e. the most reported drawbacks are at the top.

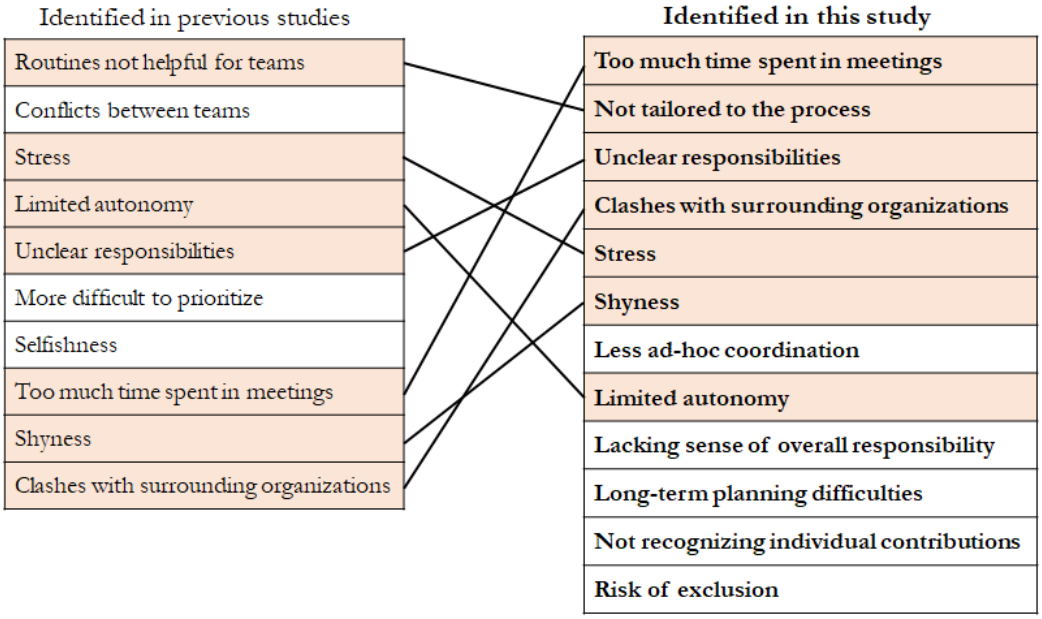

Figure 20. Drawbacks identified in previous studies vs. this study (ordered by frequency).

Something that really stands out and is contrary to previous studies is the drawback *conflicts between teams* identified in eight previous studies (see Table 2). This drawback was not identified in any of the three studied cas-

es. Rather, the contrary was acknowledged in many of the open-ended answers and interviews.

Neither was the drawback *more difficult to prioritize*, identified in four previous studies (see Table 2), seen in any of the studied cases. Instead, *clear priorities* were highlighted as perceived benefits in the studied cases. One reason for this difference might be the strong focus on visualization in the studied cases of this study. The focus on using the program board might not only have improved the view on priorities but might, in turn, also have settled potential conflicts among teams.

Lacking sense of overall responsibility and *long-term planning difficulties* were only identified at Auto, and *not recognizing individual contributions* and *risk of exclusion* were only identified at Bank. These single identified drawbacks in only one case might be context-specific, but further research is needed to investigate whether that is the case.

The inter-team coordination routines suggested by SAFe are all presented in a context where teams meet physically (Scaled Agile, 2020). For many organizations today, that is not a viable option. In this study, offices in remote locations were present in two of the cases, Auto and Gov. This did not stop them from using the prescribed routines with only minor tailoring due to the distributed setting. Scrum of Scrums was conducted via on-line meeting tools instead of meeting in the same room, for example. The program board was digital instead of physical at Auto. One might suspect that this virtual environment would cause problems, but looking at the identified drawbacks, none is related to the distributed setting. This was also in line with the SAFe implementation study at Comptel (used in the SLR), where the interviewees found the routines to be quite successful and did not mention global distribution as a problem (Paasivaara, 2017).

10.7 Perceived changes to team autonomy

One of the research questions specifically considered perceived changes to team autonomy and has been discussed in each of the case sections. A synthesis regarding similarities and differences will be presented in this section. Table 40 displays, in short, what was highlighted in each different case. It is no surprise that the perceived changes differ between the studied cases since the performance of an autonomous Agile team depends not only on the competence of the team itself, but also on the organizational context provided by management (Hoda & Noble, 2017; Stray et al., 2018).

Table 40. Case-level matrix of perceived changes to team autonomy.

	Auto	Bank	Gov
Not in accordance with Agile Principle 5 and 11*.	Cannot choose what to work with as flexibly as before	Less personal autonomy despite team autonomy	Cannot choose what to work with as flexibly as before
			Compelled to plan for 100 percent of capacity in each PI
			Approval culture
Not in accordance with the Agile Principle 10*.		Teams focus too much on their own goals instead of helping others	
Not in accordance with the Agile Principle 9*.		Architecture limitations to team autonomy	
In accordance with Agile Principle 8*	Signaling limitations *and* Clear long-term goals for the team	Better overall picture, part of a larger whole	
In accordance with Agile Principle 9*	Both receiving help and help other teams		

*** AP5:** *Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.*
AP8: *Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.*
AP9: *Continuous attention to technical excellence and good design enhances agility.*
AP10: *Simplicity - the art of maximizing the amount of work not done - is essential.*
AP 11: *The best architectures, requirements, and designs emerge from self-organizing teams.*

10.7.1 Similarities in all three cases

Two Agile principles, AP5 and AP 11 (see Table 40 above), are formulated to ensure team autonomy in ASD. In all three cases, examples of events not in accordance with these principles have been identified. A similarity between the Auto and the Gov case concerns the perceived possibility to choose what to work with in the different teams – they could not choose features as freely as before. Although the Product Owner is responsible for deciding what to build, in a team of teams setting that does not mean that the Product Owner should decide what each team does. Dividing work between teams is a part of deciding *how* to best accomplish the work (Sutherland & Schwaber, 2013).

One reason for this limited possibility of choice was that more pre-work was conducted before each planning session. The pre-work, or refinement, meant that a certain team was considered for each feature. Either they were chosen by Product Owners and product management, or the team was involved in the refinement, which, logically, was considered to be responsible for working with the feature due to more insights into the specific feature.

Another example of events perceived as not being in accordance with AP 5 and AP 11 was identified at Gov. According to informants and observations, Scrum Masters forced the team to plan for 100 percent of capacity in each PI. This happened even when coaches expressed a wish to not plan for more than 85 percent of the team's capacity. I argue that this behavior is due to the predominant *resource efficiency logic* prevailing at Gov. Since *resource efficiency logic* regards unallocated time as wasted, this logic caused Scrum Masters and managers to pressure teams into planning for full capacity. Also at Gov, a behavior not in accordance with AP 5 was an approval culture. Employees often needed to verify that decisions where approved by people higher up in the hierarchy, thereby hindering mandate to make decisions.

Another example of perceived loss of autonomy was identified at Bank where developers experienced less personal autonomy. The need for team versus individual autonomy might be in conflict, especially in teams with a high degree of diversity (Stray et al., 2018). Some employees felt that the expected transparency of always presenting performed work was a struggle. They felt forced to comply despite not wanting to. This was not in accordance with AP 5 which is to give individuals the support they need and trust them to get the job done.

10.7.2 Similarities in two of the cases

At both Auto and Bank, experiences in accordance with AP 8 were identified. The most diverse views of changes in team autonomy came from case Auto. Some teams experienced less autonomy, some thought it was the same, and some experienced more autonomy. A suggestion to why these changes happened came from John POSM, who claimed that teams with clear long-term goals had more autonomy and vice versa. Setting long-term goals is in accordance with AP 8.

With the possibility of longer planning horizons, one could argue that there is a better chance for the team to decide more on their own on how to achieve these goals. Whether or not this is true, an example of another

large-scale organization that claims the importance of a clear mission is Spotify (Šmite et al., 2020). Further research is needed to verify whether there is a causality link between long-term goals and team autonomy.

At Bank, teams expressed how they gained a better overview of their work in a larger context. This is in accordance with AP 8. And the ability to create a shared vision has been shown in research to be a key for success since the beginning of the 1990s (Gren, 2020).

10.7.3 Differences between the cases

Employees at Bank experienced two areas not in accordance with two of the Agile principles. One area was related to teams focusing too much on their own goals instead of helping others. This could sometimes even be intentional as described in Moe et al. (2019) where team members stated that they shielded their team from externalities, filtering out what they considered as unnecessary for their team to take part in. This behavior is not in accordance with AP 10, since most prioritized work might not be conducted. Although there is an outspoken goal per sprint for each team, the work of Team X, for example, might be more important to solve for the whole product rather than each team's goal.

Also at Bank, the technical architecture was considered a limitation to team autonomy, which is not in accordance with AP 9. This is a known problem, and one solution to enable team autonomy, investigated by Gundelsby (2018), is to organize the architecture by domains with APIs and service boundaries.

One perceived change to team autonomy only experienced at Auto was both the increased ability to receive help and to help other teams, which is supported by AP 9.

Neither the informants of case Auto nor of Bank perceived a difference in team autonomy during implementation of the new routines. They reported the same changes in team autonomy early in the process as well as late in the observed process of implementation. In the Gov case, however, the autonomy was reported to have changed during implementation once the Agile coaches left the train. They reported a sense of freedom and that it seemed as the Agile coaches had forced a specific way of working on the teams in accordance with an *Agile rulebook logic*, without much leeway for changes within the team. This changed once the teams were on their own. This suggests that coaches implementing large-scale coordination routines by the book, in accordance with an *Agile rulebook logic* might limit team autonomy.

11 Final discussions and conclusions

The study conducted in this thesis focuses on the phenomenon of inter-team coordination in large-scale ASD projects. To do this, I have used a multiple case study to investigate and present several similarities and differences, as well as identified patterns emerging across the three cases. As previously described, the research field could be expressed as Information Systems Development and the Project Management of Systems Development. This research area is closely related to both Software Engineering as well as Organizational Research, and research from these neighboring areas has contributed to further understanding and insights into the investigated area.

Therefore, this thesis shows that adopting theories and insights from other research domains is important for gaining further knowledge of the studied phenomenon, thereby avoiding the risk of encapsulation (Davies et al. 2018). In this chapter, I will discuss, reflect on, and summarize the findings of this study.

11.1 Tailoring inter-team coordination routines

Methods and frameworks are different in how they support processes, practices, and routines to be tailored. A method with guidance on how to tailor is what Iivari (1989) calls a method with *built-in contingency*. According to an analysis of ASD literature by Conboy and Fitzgerald (2007), it seems that Agile methods have not adequately offered this guidance. Creators of Agile methods and frameworks still offer detailed, step-by-step instructions on how to perform routines. At the same time, however, they acknowledge that tailoring needs to be conducted.

Below, I have divided my reflections and discussions on routine tailoring in this study in four different topics in order to further discuss implications of the answers to RQ1 (*How are inter-team coordination routines performed and tailored in large-scale Agile software development?*), RQ2 (*Why are inter-team coordination routines tailored (or not tailored)?*), and RQ3 (*What perceived impacts (benefits and drawbacks) are associated with the implementation of inter-team coordination routines?*).

11.1.1 The risk of a framework becoming the norm

The case study conducted at Ericsson R & D (Paasivaara et al., 2018) expressed the problems of scaling up ASD without the use of a framework. However, the study presented in this thesis shows that there are also risks

when frameworks are implemented. In particular, there is a risk for suppressing tailoring if the Agile framework becomes the norm by adhering to an *Agile rulebook logic*. I argue that this might be the case when there is too much coach support, as could be seen at Gov. Much of the discussions between coaches at Gov related to whether something was *correct* according to how it was prescribed in SAFe. Instead of trying to adapt or even invent routines on their own, coaches used SAFe as a rulebook for their implementation. At Gov, a lot less tailoring was conducted compared to the other cases, and the greatest amount of voices were raised in complaining about the implemented routines not fitting the processes.

11.1.2 The risk of falling back into old habits

Another possible risk in implementing new routines is falling back into old habits. The Agile methods and frameworks have invented new names for routines to highlight differences compared to the previous way of working. One such example is the daily team meeting, which was renamed in Scrum to *Daily Scrum* (Schwaber & Beedle, 2002). The originators explicitly chose not to use the word *meeting* for the Daily Scrum routine since they wanted to highlight that this was not a traditional meeting, such as a typical project status meeting. Instead, the focus should be on discussing and solving problems (Schwaber & Beedle, 2002).

Even though the originators stressed this important difference, many teams spend too much time on reporting progress (Stray et al., 2016), i.e. fall back into old habits of conducting traditional status meetings. For them, the new routines become old wine in new bottles where they use new names but keep the same format as before. This means that they not only implement the routines in an intended way, but also that they tailor them according to the old way of working by falling back into old habits. This was identified in both the Auto and Gov case. The Scrum of Scrums routine risks ending up with this kind of backward tailoring to be conducted as traditional status meetings.

11.1.3 Inventing new routines

The program board is used to highlight dependencies between teams during the PI planning event. According to SAFe: "it may, or may not, be maintained (manually) after planning is complete" (Scaled Agile, 2020). SAFe describes that the program board "is often used during the Scrum of Scrums to track dependencies" but does not give any guidance on how

to keep the program board up to date, only that it is optional for the organization to do so (ibid.).

At Bank, they realized that the intention of keeping the program board up to date was easier said than done. Although they did use the program board to track dependencies during Scrum of Scrums, as SAFe prescribes, they did not have time to update the board. Also, since different people attended different Scrum of Scrums, making mandated decisions on dependencies was not easy. The intention of keeping it up to date was not enough. Therefore, Bank invented a new routine to keep the program board up to date: the *mid-sprint review*. The invention of this routine arose due to lack of guidance from SAFe and the perceived importance of solving emerging dependency issues. Besides adding a useful routine to coordination, it also led to an improved way of using the program board, adding more information to the board than originally suggested by SAFe (ibid.).

11.1.4 A dynamic view of routines

Coordination routines are not stable but dynamic and are always in the making through the lifetime of a project (Moe et al., 2018). Especially in terms of coordination routines for the group mode (Van de Ven et al., 1976), this is important in large-scale ASD projects. In more practical terms, I argue, just like Moe et al. (2018), that it is not only important to be aware of how to organize a project in a start-up phase, but also to continuously evaluate and change coordination routines over time as the project progresses. New coordination routines may emerge out of the practice of project participants in a bottom-up fashion, or they can be established top-down by managers in the organization (Moe et al., 2018). Both scenarios are seen in the studied cases in this thesis.

Managers decided on implementing coordination routines described in SAFe in all three cases, but also, new coordination routines emerged in a bottom-up fashion through initiatives from team members. The routines changed over time in all cases, which was also presented in Moe et al. (2018). This dynamic view of routines is not presented in SAFe (Scaled Agile, 2020), which gives a static view of how routines are performed. Therefore, I argue based on evidence found in the three cases that it is important to keep a dynamic view of inter-team coordination routines, expecting them to change over time.

11.2 Dependencies in large-scale ASD

There are many types of dependencies in software development (Espinosa et al., 2007). Besides technical dependencies, there are other types of coordination challenges faced by software teams. Different tasks have different types of dependencies, thus requiring different types of coordination (Espinosa et al., 2007). Therefore, investigating dependency types will also, to some extent, give further answer to RQ2 (*Why are inter-team coordination routines tailored (or not tailored)?*).

From a coordination theory perspective (Malone & Crowston, 1994), all three studied cases dealt with the same types of dependencies. The dependencies analyzed were of *producer-consumer* type, where the subtypes *transfer* and *precedence* were identified. This suggests that these are the different types of dependencies that might be identified in other large-scale ASD projects as well.

Another observed pattern was that *sequential* and *pooled dependencies* were recognized (Thompson, 1967) in the three investigated cases. The third type, *reciprocal dependency*, was not. By analyzing dependencies based on Thompson's (1967) definitions, this study displays how important a definition might be. As depicted in the Gov case, the definition of what a dependency is, and thereby whether it should be visualized and managed, had an important impact. Gov decided on visualizing and managing dependencies according to the definition of a sequential dependency, but not pooled dependencies. This tailoring was due to the dominant *Agile rulebook logic* at Gov since coaches in charge of the implementation interpreted dependencies according to SAFe (Scaled Agile, 2020) only to include sequential dependencies.

Even so, most identified dependencies from observations were pooled dependencies. This made employees at Gov blind to the majority of dependencies. This, in turn, led to the view on routines to manage dependencies not fitting for the organization, since they only addressed a small part of all actual dependencies. I argue that the dominant *Agile rulebook logic* at Gov, causing only parts of dependencies to be visualized, was the reason for the negative view of the inter-team coordination routines, especially the usefulness of the program board. This view caused Gov employees to abandon the program board with subsequent problems in managing dependencies.

Another commonality between the three cases in this study was that emerging dependency issues were common and important to manage.

This data strengthens the claims of Okhuysen and Bechky (2009), that emerging dependency issues are more important to manage than what can be planned in advance, and also prevail in the domains of large-scale ASD projects. This case study also shows examples of how to manage problems caused by the emerging dependency issues. One mitigation strategy at Bank was to invent a new routine, specifically intended to solve this problem. The mid-sprint review was invented because Bank did not think the proposed routines in SAFe (Scaled Agile, 2020) were helpful enough to manage this problem. SAFe only prescribes that the program board might, or might not, be updated (ibid.). Without a supporting routine for this, Bank invented one on their own.

Another mitigation strategy for emerging dependency issues, evident both at Auto and Bank, was to increase the frequency of Scrum of Scrums and to invite more roles to these meetings. This caused decisions to be made more frequently and with more mandate due to the attending stakeholders who were Project Managers or managers from other departments. A commonality between Auto and Bank was that *flow efficiency logic* was dominating at Auto and, to some extent, at Bank. This correlation suggests a prevailing *flow efficiency logic* to be more helpful for managing emerging dependency issues, although further research is needed to investigate a possible causality.

11.3 Perceived impacts of implementing new routines

In this section, I will discuss and reflect on the perceived impacts identified in this study to further elaborate on RQ 3 (*What perceived impacts (benefits and drawbacks) were associated with the implementation of inter-team coordination routines?*) and RQ4 (*How is team autonomy perceived to be changed after implementing inter-team coordination routines?*). A systematic literature review was conducted during the work with this thesis to understand what impacts have been identified in other studies. The systematic literature review showed that only a small number of studies have focused on this topic, mainly because it is such a new phenomenon. The similarities and differences between this study and previous studies are presented and discussed below.

11.3.1 Benefits

Overview and transparency, and *coordination and cooperation* where the two most commonly reported benefits in all three cases and were also most often identified in previous studies (see Table 2). This suggests that the perceived benefits might be expected in implementations of inter-team coordination routines, regardless of context. Another benefit identified in both previous studies, and in all three cases in this study, was an increased *speed, focus, and efficiency*.

Increased *employee motivation* was identified at Auto and Bank as well as in previous studies. To speculate on the reason for improved motivation, I argue that it has to do with the most reported benefits of *overview and transparency*. Amabile and Kramer (2007) suggest that perceived progress is an important factor for perceiving increased motivation. When employees "perceive a clear path forward" (Amabile & Kramer, 2007, p. 74), motivation levels will rise. With better overview and transparency, it might be easier to identify the clear path forward in the daily work.

Increased *planning precision*, on the other hand, was not identified in any of the investigated previous studies but identified at both Auto and Bank. Implementing Agile ways of working in large-scale contexts has sometimes been accused of being inappropriate, that a more plan-driven approach would be more suitable (Hekkala et al., 2017; Turk, France, & Rumpe, 2002). The results from the current study contradict this belief and suggest that implementing inter-team coordination routines in large-scale ASD projects rather increase planning precision. The identified perceived benefit of *clear priorities* was also present in this study, but not in previous studies, and is an important finding worth highlighting.

Two more benefits identified in this study that were not presented in previous studies on large-scale ASD were *stress relief* and *no interference on teamwork*. Stress relief is not surprising to identify since, although not in a large-scale context, one study on ASD organizations showed a strong correlation to lower stress levels in self-organizing teams (Meier et al., 2018). These benefits give a view of a better working climate, despite the ongoing transformation in the organization. They also show that the fear of impaired efficiency due to change processes interfering in organizations might be overrated.

11.3.2 Drawbacks

The drawback most mentioned in all three cases was that *too much time is spent in meetings*, which made the progress of work too slow. It is, however, important to nuance this theme somewhat. I argue that this drawback might be more important at the beginning of implementing the routines and may not be of much importance later. I argue this because of the tailoring of the PI planning routine in all three cases. A similarity among them was that they all expanded the amount of team breakout time versus time for information dissemination. More time was spent on meetings that were perceived as useful for the teams as time and tailoring went by. The theme *not tailored to the work process* identified in this study further confirms the importance of tailoring of routines and the risk of implementing by the book, adhering to an *Agile rulebook logic*. The perceived drawback *unclear responsibilities* was identified several times due to a clash between new and old roles. This is also an important finding, since scaling up ASD often involves several new roles being appointed in the organization (Gustavsson, 2017).

Therefore, the three most reported perceived drawbacks, identified in all three cases, might be expected in implementations of inter-team coordination routines, regardless of context.

At Gov, several people reported *stress* as a drawback, especially due to added work by following the prescribed details of SAFe forced on them by Agile coaches adhering to an *Agile rulebook logic*. They also reported stress due to not allowing time for leeway in plans and redundancy due to *resource efficiency logic* at Gov. This suggests that *stress* could be expected in large-scale ASD settings where a combination of *Agile rulebook logic* and *resource efficiency logic* are dominant. In contrast, respondents at Auto (with a combination of *Agile toolbox logic* and *flow efficiency logic*) reported *stress relief* after implementation of inter-team coordination routines.

A drawback also identified in two previous studies (Sekitoleko et al., 2014; Wigander, 2014) is *shyness*. If people are too shy to speak up, there could be a problem since important information might be left out. This study further confirms a possible drawback due to *shyness* in large-scale ASD settings.

The reason for the drawback *less ad-hoc coordination* identified at Auto and Bank might be that with too much time spent in meetings, there might be a resistance to conduct ad-hoc meetings as well. This was not

identified in any of the previous studies and is also an important finding in this study.

One drawback identified in previous studies, but not in this study, was *conflicts between teams*. Rather, the contrary was acknowledged in many of the open-ended answers and interviews. Further research is needed to understand the contradictions between this study and previous studies, but one explanation might be the focus on visualization in the studied cases of this study. The focus on using the program board might not only have improved understanding on priorities but might, in turn, also have settled potential conflicts among teams. However, this potential reason needs to be further studied to increase our knowledge regarding these highlighted differences.

11.3.3 Team autonomy

In ASD, team autonomy means that the team should be allowed to decide *how* to best accomplish their work (Schwaber & Sutherland, 2013), such as choosing tools for automated testing. With several teams coordinating, some autonomy must, to some extent, be sacrificed in the individual team (Bass & Haxby, 2019). For example, it would not be feasible for each team to choose their own tool for automated testing in an organization.

As presented in the identified perceived drawbacks, limited autonomy was mentioned by some employees in the Bank and Gov cases. At the same time, two respondents (from Auto) did not perceive that the level of autonomy changed after new routines were implemented. Five respondents (three from Auto, two from Bank) even thought that team autonomy had increased after the implemented inter-team coordination routines.

One limitation put forth was, to use the words of the informants, that they could not choose features as freely as before. A reason, explicitly expressed in the Auto and Gov cases, was that more pre-work was conducted before each PI planning event. The pre-work, or refinement, meant that a certain team was considered in advance for each feature. Although not expressed by the informants at Auto, this pre-planning of each feature to a specific team was present in all three cases.

But, as expressed in the Auto case, there might be an improved side of team autonomy, depending on the team. For teams with clear long-term goals with the possibility of longer planning horizons, one could argue that there is a better chance for the team to decide more on their own how to achieve these goals. Hence, they experienced improved team autonomy in that sense. This might be the reason why two respondents ex-

perienced that there were no changes to team autonomy. However, a more detailed inquiry of causalities on how team autonomy changes in large-scale ASD is needed for further understanding.

Another view on changed autonomy, also expressed by informants at Auto, was the possibility to both offer and receive help. One of the perceived benefits (presented in chapter 10.5) was improved overview and transparency. With a better overview, teams might be able to see potential problems in their own plans and thereby know in advance that they should seek help. Also, by understanding possible problems for other teams, they might be able to help out in advance as well. This approved team autonomy is in line with another expressed detail at Auto that it is easier to signal your limitations. Especially when teams conduct PI planning together, a high level of transparency might reveal which teams have too much work planned.

In the Gov case, autonomy was reported to be changed during implementation once the Agile coaches left the train. With an *Agile rulebook logic* present, inter-team coordination routines from SAFe where implemented by the book. Hence, after the coaches had left, employees perceived a higher degree of team autonomy, since they did not need the approval of decisions on how to tailor routines in their teams.

11.4 Implications for theory

The case studies presented in this thesis are revelatory case studies, studies that reveal an unexplored phenomenon (Yin, 2017). Apart from the literature written by consultants and practitioners, this thesis is one of few detailed longitudinal descriptive studies of inter-team coordination in large-scale ASD.

11.4.1 New institutional theory

The theoretical propositions derived from new institutional theory proved to be helpful in explaining the results across cases. Thus, one theoretical contribution is how institutional logic ideal types can be used for analysis in large-scale ASD.

Doležel (2018) showed that the Agile manifesto (Beck et al., 2001) can be considered the emergence of an identity movement where activists create symbols and terminology for a new institution, thereby suggesting an institution on its own. Berente et al. (2015) also identified three different institutional logics due to context constraints, based on *Agile ways of working* as an institution separate from traditional software development.

This study shows how two dichotomous community order-based logics can be used for analytical purposes in large-scale ASD studies. Investigating the competing institutional logics *Agile rulebook logic* and *Agile toolbox logic* makes us understand the sensemaking behind tailoring decisions more clearly in the organizations. The different sources of legitimacy, authority and identity, along with basis of attention and strategy, indicate which dominant logic actors in each case adhered to.

Another proposition was to understand how corporate logics influenced tailoring decisions in the studied cases. Previous research (Joglekar & Ford, 2005; León & Farris, 2011; Tay, 2016) has presented how the corporate order-based logic *product development* as an institution also carried a dominant set of logics. This study shows how cases adhering to the dichotomous logics of *resource efficiency logic* and *flow efficiency logic* think and act differently in decision on tailoring inter-team coordination routines.

Durand et al. (2013) showed how the endorsement of new institutional logics could be used as a strategic resource at Google. Fitzgerald et al. (2003) also described how two competing logics might coexist in one organization. Both findings are further confirmed in this study where *flow efficiency logic* was presented as an intended strategic choice at both Bank and Gov. At Bank, investing corporate logics showed them to adhere to a hybrid logic with two competing logics coexisting within the same organization.

11.4.2 Coordination theory

Another contribution to theories on coordination within ASD is the studied dependency types. From a coordination theory perspective (Malone & Crowston, 1994), all three cases dealt with the same types of dependencies. When analyzing the dependency type according to Thompson's (1967) classification, *sequential* and *pooled dependencies* were recognized in all three cases. The third type, *reciprocal dependency*, was not. This suggests that only these two dependency types often exist in large-scale ASD.

One contribution to theories on ASD is regarding possible limitations to team autonomy. One area where autonomy seemed to be limited was regarding the freedom to choose upcoming work for the team. This limitation to autonomy may be due to more pre-planning called upon to conduct the PI planning routine. However, it seems that this perceived limitation in autonomy also has to do with the long-term vision for the team. This, however, needs to be further researched to establish whether there is a link. Another contribution of this study regarding autonomy is that

the sense of freedom could be linked to a dominant *Agile rulebook logic* where routines are implemented by the book, thereby limiting autonomy.

There is yet another contribution to coordination theory from this study. Okhuysen and Bechky (2009) claimed that too much of previous research focused on static ways to manage coordination, where dependency issues can be identified in advance and managed through good planning. This study strengthens their arguments regarding the importance of managing emerging dependency issues, at least in large-scale ASD projects.

11.4.3 Project management research

There is debate, mainly between practitioners (Agile Methodology, 2017; Meyer, 2016), regarding whether the term *project management* is problematic in ASD. The claim is that project management as a concept is hurting the Agile movement. Two arguments dominate: The first is that organizing in projects means putting new people together that have not worked together previously, instead of building stable teams (Meyer, 2016). Hence, with a project management approach, people will always work in new teams. The other argument is that project management prescribes a detailed plan for the whole project upfront (Agile Methodology, 2017). Hence, there cannot be a product backlog with detailed plans for the first upcoming work, and fewer details can be described for work further in the future if the traditional rules of project management are followed.

I argue that this view is based on misconceptions and false interpretations, and that Agile ways of working are still project management but of a different kind. In that sense, the findings in this thesis contribute to project management research.

In their theory of temporary organizations, Lundin and Söderholm (1995) refer to projects as temporary organizations. Projects are described as a way to organize people for a one-time, specific purpose. The temporary organization needs people with a mix of appropriate competences. As the authors explain, "the expectations and experiences gathered together in the team provide the basis for commitment within the team, and thus also a basis for motivation, communication and leadership" (Lundin and Söderholm, 1995, p. 442).

The Agile strategy for organizing is to create a stable environment for each team, keeping the same team members working together for as long as possible (Agile Alliance, 2001). However, in all three cases in this study, each organization had a different number of teams within their Agile Re-

lease Train between almost every PI. The temporality in these cases was not within each team, but rather the composition of teams belonging to the same Agile Release Train between each PI. I argue that a large-scale Agile development organization is a temporary organization, but instead of temporary team members, the number of teams belonging to the Agile Release Trains is temporary.

Regarding the other argument, the problem of having a detailed plan for the whole project, the question is how to define a *whole project*. Already in the first descriptions of Scrum (Schwaber & Beedle, 2002), the authors called a sprint a project, since it had a defined ending and the whole sprint should be planned in detail. In the studied cases, the defined ending is the PI. Although there is no outspoken lifetime for the Agile Release Train as such, each PI has a defined ending where the train can be rearranged, stopped or split into two, which happened at Gov. This means that the only real defined lifespan for an Agile Release Train is one PI at a time. Hence, it is a temporary organization with a detailed plan for one PI. That also means that the definition of a *whole project* in large-scale ASD is rather one PI at a time, which in the studied cases meant a period of two to four months.

Therefore, I argue that the Agile ways of working still refer to project management. I do not believe that there is a risk in using the term *project management* in Agile ways of working, since it still concerns the challenges of temporary organizing. The Agile Release Train may not be the same bird as a classical project, yet they are of the same species.

11.5 Implications for practice

This thesis presents inter-team coordination routines employed in different large-scale ASD settings. Although the purpose of this thesis did not include suggestions on normative guidance, practitioners can use the results when implementing and tailoring routines for coordination. By analyzing dominant institutional logics in an organization, a practitioner is guided by the possible benefits and drawbacks presented in the study, based on differences in institutional logics. Overall, the findings stress the importance of tailoring inter-team coordination routines, even when frameworks such as SAFe (Scaled Agile, 2020) present a static view of how routines should be performed. This is also supported in the results of the conducted survey where the cases that had the most amount of tailoring, Auto, and Bank, had the highest amount of employee satisfaction.

The study also shows the importance of how to define what a dependency really is. It is important for managers to discuss and decide what dependencies are, as the Gov case clearly showed. At Gov, by deciding on a too narrow definition, most dependencies were not visualized and managed, which caused problems with efficient coordination.

Regarding tailoring of the PI planning routine, one pattern was clearly identified: The amount of team breakout time increased during implementation. This suggests that it is more important to spend time on information-sharing in the PI planning event in the beginning of implementing inter-team coordination routines when employees do not know what is expected of them. However, as they learn, the amount of information-sharing can be decreased, and more time can be used for team breakouts.

One expected impact of conducting PI planning was that planning precision would increase. This was detected at both Auto and Bank by investigating measured progress. This suggests that even if planning precision is low in the beginning, it will likely increase. Hence, PI planning seems to be a useful routine to improve planning and forecasting in organizations.

Managers might be reluctant to implement the studied routines if they have a distributed setting with teams at different sites. This study, however, suggests that remote locations are not a major problem. At least none of the perceived drawbacks identified had to do with the distributed setting, which was prevalent at both Auto and Gov. A suggestion would be to tailor the inter-team coordination routines accordingly, i.e. use digital counterparts with a digital program board and Scrum of Scrums using online technology solutions. In most organizations, it is not feasible to travel for each PI planning or Scrum of Scrums. Also, as I am writing this, the effects of the COVID-19 pandemic show the importance of being able to work remotely.

A word of caution regarding the practical implications: Since the number of studied cases was small, these implications must be considered somewhat hypothetical. As always, when it comes to Agile ways of working, one needs to try, inspect, and adapt (Agile Alliance, 2001).

11.6 Limitations

Several limitations to this study need to be mentioned. The case study research method is the most appropriate when a *how* or *why* question is be-

ing asked (Yin, 2017). There are several important aspects to inter-team coordination in ASD organizations that could have been investigated by employing other research methods, but these were left out of this thesis. One limitation is that this study does not present any broad statistical information regarding the use, tailoring or impacts of different inter-team coordination routines in the software industry. Such data can be collected by conducting industry surveys. However, the validity of such surveys could be questioned due to the measurement bias created by the diverse understanding of the key terminology of coordination routines (Lavrakas, 2008).

One purpose of this research was to investigate the perceived impacts, not the measured impacts. However, this thesis presents observations on improvements made to the routines. Another limitation is, therefore, the limited validity of the observed cause-effect relationships between the improvements and their impacts. Based on the results presented in this thesis, further hypotheses regarding improvements to inter-team coordination in ASD can be formed, and other research methods could be used for testing hypotheses.

This research project takes a multiple case study approach. The three cases studied reflect a breadth of scope of the studied phenomenon and make it manageable. The businesses represented in the three organizations differ considerably since one is an automotive product development department, one is a business bank, and one is a government agency. The cases were selected specifically to prevent the results from being biased toward a specific business or industry and to capture data from different areas. However, if further cases with additional business logics would have been added, perhaps further insights could have been made.

11.7 Future work

The most obvious direction for future work is the continued study of inter-team coordination routines employed in large-scale ASD to find out how generalizable the findings presented in this thesis are.

11.7.1 Emerging dependencies in large-scale ASD

Okhuysen and Bechky (2009) called for further research on the coordination of emerging dependency issues. The importance of emerging dependencies has, to some extent, been investigated in this study, albeit from a qualitative aspect. One interesting area for future research would be to measure and analyze emerging dependencies quantitatively. In all

three studied cases, the number of emerging dependency issues was perceived differently by different individuals, even within the same organization. Both the amount and the impact of emerging dependency issues would, therefore, be of interest to expand our knowledge about.

11.7.2 Inter-team coordination routines based on other large-scale Agile frameworks

The cases studied in this thesis all adopted inter-team coordination routines based on prescriptions from SAFe (Scaled Agile, 2020). Although SAFe is the most adopted large-scale Agile framework (Laanti & Kettunen, 2019; VersionOne, 2020), there are others. A natural area for future research is to investigate organizations that have implemented other frameworks, such as LeSS (Larman & Vodde, 2010), DAD (Ambler & Lines, 2012), or the Spotify Model (Salameh & Bass, 2020) to see whether and how the perceived impacts differ.

11.7.3 The impact of long-term goals on team autonomy

At Auto, one of the employees expressed that he experienced a difference in team autonomy based on differing long-term goals. He claimed that his team experienced improved autonomy when his team knew what was expected of them for a longer period.

Further work is needed to verify whether there is a causal link between long-term goals and team autonomy.

References

Agile Alliance. (2001). *Agile 101: A short history of Agile*. https://www.agilealliance.org

Agile Methodology. (2017, February 15). *Want to be Agile? Stop doing "Projects"*. https://www.agilemethodology.org/want-to-be-agile-stop-doing-projects

Alqudah, M., & Razali, R. (2016). A review of scaling agile methods in large software development. *International Journal on Advanced Science, Engineering and Information Technology*, *6*(6), 828-837. https://doi.org/10.18517/ijaseit.6.6.1374

Alvesson, M. (2011). *Interpreting interviews*. SAGE Publications.

Amabile, T. M., & Kramer, S. J. (2007). Inner work life: Understanding the subtext of business performance. *Harvard Business Review*, *85*(1), 72-83.

Ambler, S. W., & Lines, M. (2012). *Disciplined Agile Delivery: A practitioner's guide to agile software delivery in the enterprise*. IBM press.

Appelo, J. (2011). *Management 3.0: Leading agile developers, developing agile leaders*. Pearson Education.

Avgerou, C. (2000). IT and organizational change: An institutionalist perspective. *IT & People, 13*(4), 234-262. https://doi.org/10.1108/09593840010359464

Babinet, E., & Ramanathan, R. (2008). Dependency management in a large agile environment. In G. Melnik, P. Kruchten, & M. Poppendieck (Eds.), *Agile 2008 Conference* (pp. 401-406). IEEE. https://doi.org/10.1109/agile.2008.58

Bakker, R. M., DeFillippi, R. J., Schwab, A., & Sydow, J. (2016). Temporary organizing: Promises, processes, problems. *Organization Studies, 37*(12), 1703-1719. https://doi.org/10.1177/0170840616655982

Bathallath, S., Smedberg, Å., & Kjellin, H. (2016). Managing project interdependencies in IT/IS project portfolios: A review of managerial issues. *International Journal of Information Systems and Project Management*, *4*(1), 67-82. https://doi.org/10.12821/ijispm040104

Bass, J. M. (2015). How product owner teams scale agile methods to large distributed enterprises. *Empirical Software Engineering, 20*(6), 1525–1557. https://doi.org/10.1007/s10664-014-9322-z

Bass, J. M., & Haxby, A. (2019). Tailoring product ownership in large-scale agile projects: Managing scale, distance, and governance. *IEEE Software, 36*(2), 58-63. https://doi.org/10.1109/ms.2018.2885524

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. http://www.agilemanifesto.org

Benestad, H. C., & Hannay, J. E. (2011). A comparison of model-based and judgment-based release planning in incremental software projects. In R. N. Taylor, H. Gall, & N. Medvidović (Eds.), *2011 33rd International Conference on Software Engineering,* ICSE (pp. 766-775). IEEE. https://doi.org/10.1145/1985793.1985901

Berente, N., Hansen, S. W., & Rosenkranz, C. (2015). Rule formation and change in information systems development: How institutional logics shape ISD practices and processes. In T. X. Bui & R. H. Sprague Jr. (Eds.), *Proceedings of the 48th Annual Hawaii International Conference on System Sciences* (pp. 5104-5113). IEEE. https://doi.org/10.1109/hicss.2015.604

Berente, N., & Yoo, Y. (2012). Institutional contradictions and loose coupling: Post-implementation of NASA's enterprise information system. *Information Systems Research, 23*(2), 376-396. https://doi.org/10.1287/isre.1110.0373

Bernstein, E. S., & Turban, S. (2018). The impact of the 'open' workspace on human collaboration. *Philosophical Transactions of the Royal Society B: Biological Sciences, 373*(1753), Article 20170239. https://doi.org/10.1098/rstb.2017.0239

Bhaskar, R. (1975). *A realist rheory of science*. Leeds books.

Bider, I. (2015). Agile research in information systems field: Analysis from knowledge transformation perspective. In M. Baptista Nunes, P. Isaías, & P. Powell (Eds.), *Proceedings of the 8th IADIS International Conference: Information Systems* (pp. 3-10). IADIS Press. http://www.iadisportal.org/digital-library/agile-research-in-information-systems-field-analysis-from-knowledge-transformation-perspective

Blichfeldt, B. S., & Eskeröd, P. (2008). Project portfolio management – There's more to it than what management enacts. *International Journal of Project Management*, *26*(4), 357-365. https://doi.org/10.1016/j.ijproman.2007.06.004

Blomquist, T., Hällgren, M., Nilsson, A., & Söderholm, A. (2010). Project-as-practice: In search of project management research that matters. *Project Management Journal, 41*(1), 5-16. https://doi.org/10.1002/pmj.20141

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *Software IEEE, 22*(5), 30–39. https://doi.org/10.1109/ms.2005.129

Boland, A., Cherry, G., & Dickson, R. (Eds.). (2017). *Doing a systematic review: A student's guide*. SAGE Publications.

Borg, M., Olsson, T., Franke, U., & Assar, S. (2018). Digitalization of Swedish government agencies - a perspective through the lens of a software development census. In *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Society* (pp. 37-46). ACM. https://doi.org/10.1145/3183428.3183434

Bowen, G. A. (2008). Naturalistic inquiry and the saturation concept: A research note. *Qualitative Research, 8*(1), 137-152. https://doi.org/10.1177/1468794107085301

Browning, T. R., & Ramasesh, R. V. (2007). A survey of activity network-based process models for managing product development projects. *Production and Operations Management, 16*(2), 217-240. https://doi.org/10.1111/j.1937-5956.2007.tb00177.x

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77-101. https://doi.org/10.1191/1478088706qp063oa

Campanelli, A. S., & Parreiras, F. S. (2015). Agile methods tailoring - A systematic literature review. *Journal of Systems and Software, 110*, 85-100. https://doi.org/10.1016/j.jss.2015.08.035

Cao, L., & Ramesh, B. (2007). Agile software development: Ad hoc practices or sound principles? *IT professional, 9*(2), 41-47. https://doi.org/10.1109/mitp.2007.27

Carlshamre, P. (2002). Release planning in market-driven software product development: Provoking an understanding. *Requirements Engineering, 7*(3), 139–151. https://doi.org/10.1007/s007660200010

Cicmil, S., & Hodgson, D. (2006). Making projects critical: An introduction. In D. Hodgson & S. Cicmil (Eds.), *Making projects critical* (pp. 1–25). Palgrave Macmillan.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer, 34*(11), 131–133. https://doi.org/10.1109/2.963450

Coelli, T. J., Rao, D. S. P., O'Donnell, C. J., & Battese, G. E. (2005). *An introduction to efficiency and productivity analysis* (2nd ed.). Springer Science & Business Media.

Cohn, M. (2007). *Advice on conducting the scrum-of-scrums meeting*. http://www.scrumalliance.org/articles/46-advice-on-conducting-the-scrum-of-scrums-meeting

Conboy, K., & Fitzgerald, B. (2007). The views of experts on the current state of agile method tailoring. In T. McMaster, D. Wastell, E. Fereneley, & J. DeGross (Eds.), *IFIP International Federation for Information Processing: Vol. 235. Organizational dynamics of technology-based innovation: Diversifying the research agenda* (pp. 217-234). Springer. https://doi.org/10.1007/978-0-387-72804-9_15

Conboy, K., & Fitzgerald, B. (2010). Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion. *ACM Transactions on Software Engineering and Methodology (TOSEM), 20*(1), Article 2. https://doi.org/10.1145/1767751.1767753

Conforto, E. C., Salum, F., Amaral, D. C., da Silva, S. L., & de Almeida, L. F. M. (2014). Can agile project management be adopted by industries other than software development? *Project Management Journal*, *45*(3), 21-34. https://doi.org/10.1002/pmj.21410

Critical Appraisal Skills Programme (2017). *CASP Qualitative Research Checklist*. http://www.casp-uk.net/casp-tools-checklists

Crowston, K., Rubleske, J., & Howison, J. (2003). Coordination theory: A ten-year retrospective. In P. Zhang & D. Galletta (Eds.), *Human-computer interaction in management information systems* (pp. 120-138). M. E. Sharpe.

Cruzes, D. S., & Dybå, T. (2011). Recommended steps for thematic synthesis in software engineering. In *2011 International Symposium on Empirical Software Engineering and Measurement. ESEM '11* (pp. 275–284). IEEE. https://doi.org/10.1109/esem.2011.36

Cruzes, D. S., Dybå, T., Runeson, P., & Höst, M. (2015). Case studies synthesis: A thematic, cross-case, and narrative synthesis worked example. *Empirical Software Engineering, 20*(6), 1634-1665. https://doi.org/10.1007/s10664-014-9326-8

Czarniawska-Joerges, B. (2007). *Shadowing and other techniques for doing fieldwork in modern societies*. Copenhagen Business School Press.

Danermark, B., Ekström, M., Jakobsen, L., & Karlsson, J. C. (2003). *Att förklara samhället*. Studentlitteratur.

Davies, A., Manning, S., & Söderlund, J. (2018). When neighboring disciplines fail to learn from each other: The case of innovation and project management research. *Research Policy, 47*(5), 965-979. https://doi.org/10.1016/j.respol.2018.03.002

Diefenbach, T. (2009). Are case studies more than sophisticated storytelling? Methodological problems of qualitative empirical research mainly based on semi-structured interviews. *Quality and Quantity, 43*(6), 875–894. https://doi.org/10.1007/s11135-008-9164-0

Dietrich, P., Kujala, J., & Artto, K. (2013). Inter-team coordination patterns and outcomes in multi-team projects. *Project Management Journal, 44*(6), 6-19. https://doi.org/10.1002/pmj.21377

Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software, 119*, 87-108. https://doi.org/10.1016/j.jss.2016.06.013

Dingsøyr T., Fægri T. E., & Itkonen, J. (2014). What is large in large-scale? *A taxonomy of scale for agile software development*. In A. Jedlitschka, P. Kuvaja, M. Kuhrmann, T. Männistö, J. Münch, & M. Raatikainen (Eds.), *Lecture notes in computer science: Vol. 8892. Product-focused software process improvement, PROFES 2014* (pp. 273-276). Springer. https://doi.org/10.1007/978-3-319-13835-0_20

Dingsøyr, T., & Moe, N. B. (2014). Towards principles of large-scale agile development. In T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, & K. Petersen (Eds.), *Lecture notes in business information processing: Vol. 199. Agile methods: Large-Scale development, refactoring, testing, and estimation* (pp. 1-8). Springer. https://doi.org/10.1007/978-3-319-14358-3_1

Dingsøyr, T., Moe, N. B., Fægri, T. E., & Seim, E. A. (2018). Exploring software development at the very large-scale: A revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering, 23*(1), 490-520. https://doi.org/10.1007/s10664-017-9524-2

Dingsøyr, T., Moe, N. B., & Seim, E. A. (2018). Coordinating knowledge work in multi-team programs: Findings from a large-scale agile development program. *Project Management Journal, 49*(6), 64-77. https://doi.org/10.1177/8756972818798980

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software, 85*(6), 1213–1221. https://doi.org/10.1016/j.jss.2012.02.033

Doležel M. (2018) Possibilities of applying institutional theory in the study of hybrid software development concepts and practices. In M. Kuhrmann (Ed.), *Lecture notes in computer science: Vol. 1127. Product-focused software process improvement* (pp. 441-448). Springer. https://doi.org/10.1007/978-3-030-03673-7_35

Donaldson, L. (2001). *The contingency theory of organizations*. SAGE Publications.

Durand, R., Szostak, B., Jourdan, J., & Thornton, P. H. (2013). Institutional logics as strategic resources. *Research in the Sociology of Organizations, 39*(Part A), 165-201.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology, 50*(9-10), 833-859. https://doi.org/10.1016/j.infsof.2008.01.006

Ebert, C., & Paasivaara, M. (2017). Scaling agile. *IEEE Software, 34*(6), 98-103. https://doi.org/10.1109/ms.2017.4121226

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review, 14*(4), 532-550. https://doi.org/10.2307/258557

Elshamy, A., & Elssamadisy, A. (2006). Divide after you conquer: An agile software development practice for large projects. In P. Abrahamsson, M. Marchesi, & G. Succi (Eds.), *Lecture notes in computer science: Vol. 4044. Extreme programming and agile processes in software engineering* (pp. 164-168). Springer. https://doi.org/10.1007/11774129_17

Emerson, R. M., Fretz, R. I., & Shaw, L. L. (2011). *Writing ethnographic fieldnotes*. University of Chicago Press.

Engwall, M. (2003). No project is an island: Linking projects to history and context. *Research Policy, 32*(5), 789-808. https://doi.org/10.1016/s0048-7333(02)00088-4

Espinosa, J. A., Slaughter, S. A., Kraut, R. E., & Herbsleb, J. D. (2007). Team knowledge and coordination in geographically distributed software development. *Journal of Management Information Systems, 24*(1), 135-169. https://doi.org/10.2753/MIS0742-1222240104

Etcheverry, P., Lopistéguy, P., & Dagorret, P. (2001a). Pattern-based guidelines for coordination engineering. In H. C. Mayr, J. Lazansky, G. Quirchmayr, & P. Vogel (Eds.), *Lecture notes in computer science: Vol 2113. Database and expert systems applications, DEXA* (pp. 155-164). Springer. https://doi.org/10.1007/3-540-44759-8_17

Etcheverry, P., Lopistéguy, P., Dagorret, P. (2001b). Specifying contexts for coordination patterns. In V. Akman, P. Bouquet, R. Thomason, & R. Young (Eds.), *Lecture notes in computer science: Vol. 2116. Modeling and using context* (pp. 437-440). Springer. https://doi.org/10.1007/3-540-44607-9_38

Evbota, F., Knauss E., Sandberg A. (2016) Scaling up the planning game: Collaboration challenges in large-scale agile product development. In H. Sharp & T. Hall (Eds.), *Lecture notes in business information processing: Vol. 251. Agile processes, in software engineering, and extreme programming* (pp. 28-38). Springer. https://doi.org/10.1007/978-3-319-33515-5_3

Feldman, M. S., & Pentland, B. T. (2003). Reconceptualizing organizational routines as a source of flexibility and change. *Administrative Science Quarterly, 48*(1), 94–118. https://doi.org/10.2307/3556620

Ferrer Conill, R. (2018). *Gamifying the news: Exploring the introduction of game elements into digital journalism* [Doctoral dissertation, Karlstad University]. DIVA. http://www.diva-portal.org/smash/get/diva2:1240298/FULLTEXT04.pdf

Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems, 15*(2), 200-213. https://doi.org/10.1057/palgrave.ejis.3000605

Fitzgerald, B., Russo, N. L., & O'Kane, T. (2003). Software development method tailoring at Motorola. *Communications of the ACM, 46*(4), 64-70. https://doi.org/10.1145/641205.641206

Flyvbjerg, B. (2011). Over budget, over time, over and over again: Managing major projects. In P. W. G. Morris, J. K. Pinto, & J. Söderlund (Eds.), *The Oxford handbook of project management* (pp. 321–344). Oxford University Press.

Flyvbjerg, B. (2018). Planning fallacy or hiding hand: Which is the better explanation? *World Development, 103*, 383–386. https://doi.org/10.1016/j.worlddev.2017.10.002

Fowler, M. (2000). Put your process on a diet. *Software Development, 8*(12), 32-36.

Friedland, R., & Alford, R. R. (1991). Bringing society back in: Symbols, practices and institutional contradictions. In W. W. Powell & P. DiMaggio (Eds.), *The new institutionalism in organizational analysis* (pp. 232–263). University of Chicago Press.

Frith, H., & Gleeson, K. (2004). Clothing and embodiment: Men managing body image and appearance. *Psychology of Men & Masculinity, 5*(1), 40-48.

https://doi.org/10.1037/1524-9220.5.1.40

Galbraith, J. (1973). *Designing complex organizations*. Addison-Wesley.

Gold, R. (1958). Roles in sociological field observations. *Social Forces, 36*(3), 217-223. https://doi.org/10.2307/2573808

Goldkuhl, G. (2012). From action research to practice research. *Australasian Journal of Information Systems, 17*(2), 57-78. https://doi.org/10.3127/ajis.v17i2.688

Gregory P., Barroca L., Taylor K., Salah D., & Sharp H. (2015). Agile challenges in practice: A thematic analysis. In C. Lassenius, T. Dingsøyr, & M. Paasivaara (Eds), *Lecture notes in business information processing: Vol. 212. Agile processes in software engineering and extreme programming* (pp. 64-80). Springer. https://doi.org/10.1007/978-3-319-18612-2_6

Gren, L. (2020). Understanding work practices of autonomous agile teams: A social-psychological review. In M. Paasivaara & P. Kruchten (Eds.), *Lecture notes in business information processing: Vol. 396. Agile processes in software engineering and extreme programming – workshops* (pp. 227-235). https://doi.org/10.1007/978-3-030-58858-8_23

Groth, L. (1999). *Future organizational design: The scope for the IT-based enterprise*. John Wiley & Sons.

Gundelsby, J. H. (2018). Enabling autonomous teams in large-scale agile through architectural principles. In S. Wagner, R. Hoda, & A. Aguiar (Eds.), *XP '18*: *Proceedings of the 19th International Conference on Agile Software Development: Companion* (Article 17). ACM. https://doi.org/10.1145/3234152.3234183

Gupta, R. K., Jain, S., & Singh, B. (2018). Challenges in scaling up a globally distributed legacy product: A case study of a matrix organization. In *ICGSE '18: Proceedings of the 13th International Conference on Global Software Engineering* (pp. 72-76). ACM. https://doi.org/10.1145/3196369.3196389

Gupta, R. K., & Reddy, P. M. (2016). Adapting agile in a globally distributed software development. In T. X. Bui & R. H. Sprague, Jr. (Eds.), *2016 Proceedings of the 49th Annual Hawaii International Conference on System Sciences* (pp. 5360-5367). IEEE. https://doi.org/10.1109/hicss.2016.663

Gustavsson, T. (2007). *Agile - konsten att slutföra projekt*. TUK Förlag.

Gustavsson, T. (2011). *Agil projektledning*. Sanoma utbildning.

Gustavsson, T. (2016). Benefits of agile project management in a non-software development context: A literature review. In *Fifth International Scientific Conference on Project Management in the Baltic Countries* (pp. 114-124). University of Latvia. https://www.econstor.eu/handle/10419/197312

Gustavsson, T. (2017). Assigned roles for inter-team coordination in large-scale agile development: A literature review. In *XP '17 Workshops: Proceedings of the Scientific Workshops of XP2017* (Article 15). https://doi.org/10.1145/3120459.3120475

Gustavsson, T. (2018). Practices for vertical and horizontal coordination in the Scaled Agile Framework. In B. Andersson, B. Johansson, S. Carlsson, C. Barry, M. Lang, H. Linger, & C. Schneider (Eds.), *Information Systems Development: Designing Digitalization, ISD2018 Proceedings*. Lund University.

https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1239&context=isd2014

Gustavsson, T. (2019a). Changes over time in a planned inter-team coordination routine. In R. Hoda (Ed.), ***Lecture notes in business information processing: Vol. 364. Agile processes in software engineering and extreme programming - workshops*** **(pp. 105-111). Springer. https://doi.org/10.1007/978-3-030-30126-2_13**

Gustavsson, T. (2019b). Dynamics of inter-team coordination routines in large-scale agile software development. In ***Proceedings of the 27th European Conference on Information Systems, ECIS*** **(pp. 1-15). Stockholm University; Uppsala University. https://aisel.aisnet.org/ecis2019_rp/40**

Gustavsson, T. (2019c). Voices from the teams - impacts on autonomy in large-scale agile software development settings. In R. Hoda (Ed.), ***Lecture notes in business information processing: Vol. 364. Agile processes in software engineering and extreme programming – workshops*** **(pp. 29-36). Springer. https://doi.org/10.1007/978-3-030-30126-2_4**

Gustavsson, T. (2020). Visualizing inter-team coordination. In ***Proceedings of EASE 2020: Evaluation and Assessment in Software Engineering*** **(pp. 306-311). ACM. https://doi.org/10.1145/3383219.3383253**

Gustavsson, T., & Bergkvist, L. (2019). Perceived impacts of using the Scaled Agile Framework for large-scale agile software development. In A. Siarheyeva, C. Barry, M. Lang, H. Linger, & C. Schneider (Eds.), ***Information Systems Development: Information Systems Beyond 2020, ISD2019 Proceedings.*** **ISEN Yncréa Méditerranée. https://aisel.aisnet.org/isd2014/proceedings2019/ManagingISD/11/**

Gustavsson, T., & Görling, S. (2019). ***Att arbeta med systemutveckling*****. Studentlitteratur.**

Gustavsson, T. & Rönnlund, P. (2010, November 1-3). ***Agile project management in public events*** **[Paper presentation]. 24th IPMA World Congress, Istanbul, Turkey.**

Gustavsson, T. & Rönnlund, P. (2013, August 21-23). ***Agile adoption at Ericsson hardware product development*** **[Paper presentation]. 22nd Nordic Academy of Management Conference, University of Iceland, Reykjavik, Iceland.**

Hekkala, R., Stein, M. K., Rossi, M., & Smolander, K. (2017). Challenges in transitioning to an agile way of working. In T. X. Bui (Ed.), *Proceedings of the 50th Hawaii International Conference on System Sciences* (pp. 5869-5878). University of Hawaii. https://doi.org/10.24251/hicss.2017.707

Heikkilä, V. T. (2015). *Case studies on release planning in agile development organizations* [Doctoral dissertation, Aalto University]. Aaltodoc. https://aaltodoc.aalto.fi/bitstream/handle/123456789/14961/isbn9789526060460.pdf?sequence=1&isAllowed=y

Hobbs, B., & Petit, Y. (2017). Agile methods on large projects in large organizations. *Project Management Journal*, *48*(3), 3-19. https://doi.org/10.1177/875697281704800301

Hoda, R., & Noble, J. (2017, May). Becoming agile: A grounded theory of agile transitions in practice. In *2017 IEEE/ACM 39th International Conference on Software Engineering, ICSE* (pp. 141-151). IEEE. https://doi.org/10.1109/icse.2017.21

Hoda, R., Noble, J., & Marshall, S. (2012). Developing a grounded theory to explain the practices of self-organizing agile teams. *Empirical Software Engineering, 17*(6), 609-639. https://doi.org/10.1007/s10664-011-9161-0

Hoda, R., & Murugesan, L. K. (2016). Multi-level agile project management challenges: A self-organizing team perspective. *Journal of Systems and Software, 117*, 245-257. https://doi.org/10.1016/j.jss.2016.02.049

Hoegl, M., Weinkauf, K., & Gemuenden, H. G. (2004). Interteam coordination, project commitment, and teamwork in multiteam R&D projects: A longitudinal study. *Organization Science*, *15*(1), 38-55. https://doi.org/10.1287/orsc.1030.0053

Hällgren, M., & Söderholm, A. (2011). Projects-as-practice: New approach, new insights. In P. W. G. Morris, J. K. Pinto, & J. Söderlund (Eds.), *The Oxford handbook of project management* (pp. 500–518). Oxford University Press.

Iivari, J. (1989). A methodology for IS development as organizational change: A pragmatic contingency approach. In H. K. Klein and K. Kumar (Eds.), *Systems development for human progress* (pp. 197-217). North-Holland.

Jansson, T. (2015). *Agila projektledningsmetoder och motivation: Varför man blir produktiv av att flytta lappar på en whiteboard* [Doctoral dissertation, Karlstad University]. DIVA. http://www.diva-portal.org/smash/get/diva2:782985/FULLTEXT01.pdf

Janz, B. D., Colquitt, J. A., & Noe, R. A. (1997). Knowledge worker team effectiveness: The role of autonomy, interdependence, team development, and contextual support variables. *Personnel Psychology, 50*(4), 877-904. https://doi.org/10.1111/j.1744-6570.1997.tb01486.x

Jarzabkowski, P. A., Lê, J. K., & Feldman, M. S. (2012). Toward a theory of coordinating: Creating coordinating mechanisms in practice. *Organization Science*, *23*(4), 907-927. https://doi.org/10.1287/orsc.1110.0693

Jeffries, R., Anderson, A., & Hendrickson, C. (2001). *Extreme programming installed*. Addison-Wesley Professional.

Jepperson, R. L. (1991). Institutions, institutional effects, and institutionalism. In W. W. Powell & P. J. DiMaggio (Eds.), *The new institutionalism in organizational analysis* (pp. 143–163). University of Chicago Press.

Joglekar, N. R., & Ford, D. N. (2005). Product development resource allocation with foresight. *European Journal of Operational Research, 160*(1), 72-87. https://doi.org/10.1016/j.ejor.2003.06.021

Kaplan, A. (1964). *The conduct of inquiry, methodology for behavioral science*. Chandler Publishing.

Karlsson, F., & Ågerfalk, P. J. (2009). Exploring agile values in method configuration. *European Journal of Information Systems, 18*(4), 300-316. https://doi.org/10.1057/ejis.2009.20

Karlström, D., & Runeson, P. (2005). Combining agile methods with stage-gate project management. *IEEE Software, 22*(3), 43-49. https://doi.org/10.1109/ms.2005.59

Karrbom Gustavsson, T., & Hallin, A. (2014). Rethinking dichotomization: A critical perspective on the use of "hard" and "soft" in project management research. *International Journal of Project Management, 32*(4), 568-577. https://doi.org/10.1016/j.ijproman.2013.10.009

Kettunen, P. (2009). *Agile software development in large-scale new product development organization: Team-level perspective.* [Doctoral dissertation, Helsinki University of Technology]. Aaltodoc. https://aaltodoc.aalto.fi/bitstream/handle/123456789/4694/isbn9789522481146.pdf?sequence=1&isAllowed=y

Kirkman, B. L., & Rosen, B. (1999). Beyond self-management: Antecedents and consequences of team empowerment. *Academy of Management Journal, 42*(1), 58-74. https://doi.org/10.5465/256874

Kitchenham, B. A. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Technical report EBSE-2007-01). Keele University.

Klünder, J., Hohl, P., & Schneider, K. (2018). *Becoming agile while preserving software product lines: An agile transformation model for large companies*. In M. Kuhrman, R. V. O´Connor, D. Houston, & R. Hebig (Eds.), *ICSSP '18: International Conference on the Software and Systems Process* (pp. 1-10) ACM. https://doi.org/10.1145/3202710.3203146

Koehnemann, H., & Coats, M. (2009). Experiences applying agile practices to large systems. In Y. Dubinsky, T. Dybå, S. Adolph, & A. Sidky (Eds.), *Agile 2009 Conference* (pp. 295-300). IEEE. https://doi.org/10.1109/agile.2009.59

Kraut, R. E., & Streeter, L. A. (1995). Coordination in software development. *Communications of the ACM*, *38*(3), 69-82. https://doi.org/10.1145/203330.203345

Kreiner, K. (2020). Conflicting notions of a project: The battle between Albert O. Hirschman and Bent Flyvbjerg. *Project Management Journal*, *51*(4), 400-410. https://doi.org/10.1177/8756972820930535

Laanti, M. (2012). *Agile methods in large-scale software development organizations: Applicability and model for adoption* (Doctoral thesis, University of Oulu). Jultika. http://jultika.oulu.fi/files/isbn9789526200347.pdf

Laanti, M., & Kettunen, P. (2019, May). SAFe adoptions in Finland: A survey research. In R. Hoda (Ed.), *Lecture notes in business information processing: Vol. 364. Agile processes in software engineering and extreme programming – workshops* (pp. 81-87). Springer. https://doi.org/10.1007/978-3-030-30126-2_10

Laanti, M., Salo, O., & Abrahamsson, P. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology, 53*(3), 276-290. https://doi.org/10.1016/j.infsof.2010.11.010

LaDonna, K. A., Taylor, T., & Lingard, L. (2018). Why open-ended survey questions are unlikely to support rigorous qualitative insights. *Academic Medicine, 93*(3), 347-349. https://doi.org/10.1097/acm.0000000000002088

Lagerberg, L., Skude, T., Emanuelsson, P., Sandahl, K., & Stahl, D. (2013). The impact of agile principles and practices on large-scale software development projects: A multiple-case study of two projects at Ericsson. In *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM* (pp. 348-356). IEEE. https://doi.org/10.1109/esem.2013.53

Larman, C. (2004). *Agile and iterative development: A manager's guide.* Addison-Wesley.

Larman, C., & Vodde, B. (2010). *Practices for scaling lean and agile development: Large, multi-site and offshore product development with large-scale Scrum*. Addison Wesley.

Larsson, P. (2008, October 22). *Kritiken mot Scrum växer*. Computer Sweden. https://computersweden.idg.se/2.2683/1.187182/kritiken-mot-scrum-vaxer

Lavrakas, P. J. (2008). *Encyclopedia of survey research methods*. SAGE Publications.

Leffingwell, D. (2007). *Scaling software agility: Best practices for large enterprises*. Addison Wesley.

Leffingwell, D., Yakyama, A., Knaster, R., Jemilo, D., & Oren, I. (2017). *SAFe reference guide: Scaled Agile Framework for lean software and systems engineering*. Addison Wesley.

Leigh Star, S. (2010). This is not a boundary object: Reflections on the origin of a concept. *Science, Technology, & Human Values. 35*(5), 601-617. https://doi.org/10.1177/0162243910377624

León, H. C. M., & Farris, J. A. (2011). Lean product development research: Current state and future directions. *Engineering Management Journal, 23*(1), 29-51. https://doi.org/10.1080/10429247.2011.11431885

Licorish, S. A., Holvitie, J., Hyrynsalmi, S., Leppänen, V., Spínola, R. O., Mendes, T. S., MacDonell, S. G., & Buchan, J. (2016). Adoption and suitability of software development methods and practices. In A. Potanin, G. C. Murphy, S. Reeves, & J. Dietrich (Eds.), *2016 23rd Asia-Pacific Software Engineering Conference, APSEC* (pp. 369-372). IEEE. https://doi.org/10.1109/apsec.2016.062

Lundin, R. A., & Söderholm, A. (1995). A theory of the temporary organization. *Scandinavian Journal of Management, 11*(4), 437-455. https://doi.org/10.1016/0956-5221(95)00036-u

Machado, M. A. (2013). New product development: From efficiency to value creation. In *2013 Proceedings of PICMET'13: Technology Management in the IT-Driven Services* (pp. 1542-1549). IEEE.

Maples, C. (2009, August). Enterprise agile transformation: The two-year wall. In Y. Dubinsky, T. Dybå, S. Adolph, & A. Sidky (Eds.), *2009 Agile Conference* (pp. 90-95). IEEE. https://doi.org/10.1109/agile.2009.62

March, J. G., & Simon, H. A. (1958). *Organizations*. John Wiley and Sons.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems, 15*(4), 251-266. https://doi.org/10.1016/0167-9236(94)00041-2

Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys*, *26*(1), 87-119. https://doi.org/10.1145/174666.174668

Manning, S. (2008). Embedding projects in multiple contexts – a structuration perspective. *International Journal of Project Management, 26*(1), 30-37. https://doi.org/10.1016/j.ijproman.2007.08.012

Mathieu, J., Maynard, M. T., Rapp, T., & Gilson, L. (2008). Team effectiveness 1997-2007: A review of recent advancements and a glimpse into the future. *Journal of Management, 34*(3), 410-476. https://doi.org/10.1177/0149206308316061

Maximini, D. (2013, September 11). A critical view on SAFe. Scrumorakel.de/blog. http://scrumorakel.de/blog/index.php?/archives/45-A-critical-view-on-SAFe.html

McBreen, P. (2003). *Questioning extreme programming*. Addison-Wesley.

McCall, M. W., Morrison, A. M., & Hannan, R. L. (1978). *Studies of managerial work: Results and methods*. Center for Creative Leadership.

McHugh, O., Conboy, K., & Lang, M. (2011). Agile practices: The impact on trust in software project teams. *IEEE Software, 29*(3), 71-76. https://doi.org/10.1109/ms.2011.118

Meier A., Kropp M., Anslow C., Biddle R. (2018). Stress in agile software development: Practices and outcomes. In J. Garbajosa, X. Wang, A. Aguiar (Eds.), *Lecture notes in business information processing: Vol. 314. Agile processes in software engineering and extreme programming* (pp. 259-266). Springer. https://doi.org/10.1007/978-3-319-91602-6_18

Merriam, S. B. (1988). *Case study research in education: A qualitative approach*. Jossey-Bass Publishers.

Meyer, W. (2016). *Is agile project management, or is it something different.* https://www.linkedin.com/pulse/agile-project-management-something-different-werner-meyer-phd/

Miles, M. B., Huberman, A. M., & Saldaña, J. (2014). *Qualitative data analysis: A methods sourcebook* (3rd ed.). SAGE Publications.

Mintzberg, H. (1983). *Structures in fives*. Prentice-Hall.

Mintzberg, H. (1998). Five Ps for strategy. In H. Mintzberg, J.B. Quinn and S. Ghoshal (Eds.)*, The strategy process* (pp. 10-17). Prentice Hall.

Moe, N. B., Dahl, B., Stray, V., Karlsen, L. S., & Schjødt-Osmo, S. (2019). Team autonomy in large-scale agile. In T. X. Bui (Ed.), *Proceedings of the 52nd Hawaii International Conference on System Sciences* (pp. 6997-7006). University of Hawaii. https://doi.org/10.24251/hicss.2019.839

Moe, N. B., Dingsøyr, T., & Dybå, T. (2009). Overcoming barriers to self-management in software teams. *IEEE Software, 26*(6), 20-26. https://doi.org/10.1109/ms.2009.182

Moe, N. B., Dingsøyr, T., & Rolland, K. (2018). To schedule or not to schedule? An investigation of meetings as an inter-team coordination mechanism in large-scale agile software development. *International Journal of Information Systems and Project Management, 6*(3), 45-59. https://doi.org/10.12821/ijispm060303

Moe, N. B., Holmström Olsson, H., & Dingsøyr T. (2016). Trends in large-scale agile development: A summary of the 4th workshop at XP2016. In *XP '16 Workshops: Scientific Workshop Proceedings of XP2016* (Article 1). ACM. https://doi.org/10.1145/2962695.2962696

Moe, N. B., Stray, V., & Hoda, R. (2019). Trends and updated research agenda for autonomous agile teams: A summary of the second international workshop at XP2019. In R. Hoda (Ed.)*, Lecture notes in business information processing: Vol. 364. Agile processes in software engineering and extreme programming – workshops* (pp. 13-19). Springer. https://doi.org/10.1007/978-3-030-30126-2_2

Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (2009). Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *Annals of Internal Medicine, 151*(4), 264-269. https://doi.org/10.7326/0003-4819-151-4-200908180-00135

National Institute for Health Research. (2011). *PROSPERO, International prospective register of systematic reviews*. http://www.crd.york.ac.uk/prospero/

North, D. (1990). *Institutions, institutional change and economic performance*. Cambridge University Press.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72-78. https://doi.org/10.1145/1060710.1060712

Nuottila, J., Aaltonen, K., & Kujala, J. (2016). Challenges of adopting agile methods in a public organization. *International Journal of Information Systems and Project Management, 4*(3), 65-85. https://ijispm.sciencesphere.org/archive/ijispm-040304.pdf

Nyrud, H., & Stray, V. (2017). Inter-team coordination mechanisms in large-scale agile. In *XP '17 workshops: Proceedings of the scientific Workshops of XP2017* (Article 16). ACM. https://doi.org/10.1145/3120459.3120476

O'Mahony, S., & Lakhani, K. R. (2011). Organizations in the shadow of communities. In C. Marquis, M. Lounsbury, & R. Greenwood (Eds.), *Communities and Organizations* (pp. 3-36). Emerald.

Ocasio, W., & Gai, S. L. (2020). Institutions: Everywhere but not everything. *Journal of Management Inquiry*, *29*(3), 262-271. https://doi.org/10.1177/1056492619899331

Ocasio, W., Thornton, P. H., & Lounsbury, M. (2017). The institutional logics perspective. In R. C. Greenwood, C. Oliver, T. B. Lawrence, & R. E. Meyer (Eds.), *The SAGE handbook of organizational institutionalism* (pp. 599–531). SAGE Publications.

Okhuysen, G. A., & Bechky, B. A. (2009). 10 coordination in organizations: An integrative perspective. *Academy of Management Annals, 3*(1), 463-502. https://doi.org/10.5465/19416520903047533

Okoli, C. (2015, December 8). Critical realist considerations for literature reviews. *SSRN*. https://doi.org/10.2139/ssrn.2700524

Olszewska, M., Heidenberg, J., Weijola, M., Mikkonen, K., & Porres, I. (2016). Quantitatively measuring a large-scale agile transformation. *Journal of Systems and Software, 117*, 258-273. https://doi.org/10.1016/j.jss.2016.03.029

Orlikowski, W. J., & Barley, S. R. (2001). Technology and institutions: What can research on information technology and research on organizations learn from each other? *MIS quarterly, 25*(2), 145-165. https://doi.org/10.2307/3250927

Paasivaara, M. (2017). Adopting SAFe to scale agile in a globally distributed organization. In *12th International Conference on Global Software Engineering* (pp. 36-40). IEEE. https://doi.org/10.1109/icgse.2017.15

Paasivaara, M., Behm, B., Lassenius, C., & Hallikainen, M. (2014). Towards rapid releases in large-scale XaaS development at Ericsson: A case study. In *9th International Conference on Global Software Engineering* (pp. 16–25). IEEE. https://doi.org/10.1109/icgse.2014.22

Paasivaara, M., Behm, B., Lassenius, C., & Hallikainen, M. (2018). Large-scale agile transformation at Ericsson: a case study. *Empirical Software Engineering, 23*(5), 2550-2596. https://doi.org/10.1007/s10664-017-9555-8

Paasivaara, M., & Lassenius, C. (2016). Scaling Scrum in a large globally distributed organization: A case study. In *IEEE 11th International Conference on Global Software Engineering, ICGSE* (pp. 74-83). IEEE. https://doi.org/10.1109/icgse.2016.34

Paasivaara, M., Lassenius, C., & Heikkilä, V. T. (2012). Inter-team coordination in large-scale globally distributed Scrum: Do Scrum-of-Scrums really work? *Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM* (pp. 235-238). IEEE. https://doi.org/10.1145/2372251.2372294

Packendorff, J. (1995). Inquiring into the temporary organization: New directions for project management research. *Scandinavian Journal of Management, 11*(4), 319-333. https://doi.org/10.1016/0956-5221(95)00018-q

Patton, M. Q. (2002). *Qualitative research and evaluation methods* (3rd ed.). SAGE Publication.

Pentland, B. T. (2003). Conceptualizing and measuring variety in the execution of organizational work processes. *Management Science, 49*(7), 857–870. https://doi.org/10.1287/mnsc.49.7.857.16382

Pentland, B. T., & Feldman, M. S. (2005). Organizational routines as unit of analysis. *Industrial and Corporate Change, 14*(5), 793–815. https://doi.org/10.1093/icc/dth070

Pentland, B. T., Feldman, M. S., Becker, M. C., & Liu, P. (2012). Dynamics of organizational routines: A generative model. *Journal of Management Studies, 49*(8), 1484–1508. https://doi.org/10.1111/j.1467-6486.2012.01064.x

Pope, C., Mays, N., & Popay, J. (2007). *Synthesizing qualitative and quantitative health evidence: A guide to methods*. Open University Press.

Pries-Heje, J., & Krohn, M. M. (2017). The SAFe way to the agile organization. In *XP '17: Proceedings of the XP2017 Scientific Workshops* (Article 18). ACM. https://doi.org/10.1145/3120459.3120478

Prosci. (2020). *Stop confusing agile with Agile.* https://www.prosci.com/resources/articles/stop-confusing-agile-with-Agile

Reinertsten, D. G. (2009). *The principles of product development flow: Second generation lean product development*. Celeritas.

Rolland, K. H., Mikkelsen, V., & Næss, A. (2016). Tailoring agile in the large: Experience and reflections from a large-scale agile software development project. In H. Sharp & T. Hall (Eds.), *Lecture notes in business information processing: Vol. 251. Agile processes, in software engineering, and extreme programming* (pp. 244-251). Springer. https://doi.org/10.1007/978-3-319-33515-5_22

Rowlands, B. (2008). Institutional aspects of systems development. In A. Mills & S. Huff (Eds.), *Proceedings of the 19th Australasian Conference on Information Systems, ACIS 2008* (pp. 55-65). University of Canterbury. https://aisel.aisnet.org/acis2008/42

Šāblis, A., Šmite, D., & Moe, N. B. (2020). Team-external coordination in large-scale software development projects. *Journal of Software: Evolution and Process*. Advance online publication. https://doi.org/10.1002/smr.2297

Sahlin-Andersson, K., & Söderholm, A. (2002). *Beyond project management: New perspective on the temporary - permanent dilemma*. Liber.

Salameh, A., & Bass, J. (2019). Spotify tailoring for B2B product development. In M. Staron, R. Capilla, & A. Skavhaug (Eds.), *45th Euromicro Conference on Software Engineering and Advanced Applications, SEAA* (pp. 61-65). IEEE. https://doi.org/10.1109/seaa.2019.00018

Salameh, A., & Bass, J. (2020). Heterogeneous tailoring approach using the Spotify model. In *Proceedings of EASE 2020: Evaluation and Assessment in Software Engineering* (pp. 293–298) ACM. https://doi.org/10.1145/3383219.3383251

Sarpiri, M. N., & Gandomani, T. J. (2017). How agile managers affect the process of software development? *IJCSNS International Journal of Computer Science and Network Security*, *17*(5), 283-286. http://paper.ijcsns.org/07_book/201705/20170538.pdf

Scaled Agile Inc. (2016). *Scaled Agile Framework 4.0*. http://v4.scaledagileframework.org

Scaled Agile Inc. (2020). *Scaled Agile Framework 5.0*. http://www.scaledagileframework.org

Scheerer, A., Hildenbrand, T., & Kude, T. (2014). *Coordination in large-scale agile software development: A multiteam systems perspective.* In R. H. Sprague, Jr. (Ed.), *Proceedings of the 47th Annual Hawaii International Conference on System Sciences* (pp. 4780-4788). IEEE. https://doi.org/10.1109/hicss.2014.587

Schnitter, J., & Mackert, O. (2010). Large-scale agile software development at SAP AG. In L. A. Maciaszek & P. Loucopoulos (Eds.), *Communications in computer and information science: Vol. 230. Evaluation of novel approaches to software engineering* (pp. 209-220). Springer. https://doi.org/10.1007/978-3-642-23391-3_15

Schwaber, K. (2004). *Agile project management with Scrum*. Microsoft press.

Schwaber, K. (2006). *Scrum is hard and disruptive.* http://static1.1.sqspcdn.com/static/f/447037/6486484/1270930467650/Scrum+Is+Hard+and+Disruptive.pdf

Schwaber, K. (2013). *UnSAFe at any speed.* http://kenschwaber.wordpress.com/2013/08/06/unsafe-at-any-speed/

Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum*. Prentice Hall.

Shastri, Y., Hoda, R., & Amor, R. (2017). Understanding the roles of the manager in agile project management. In *Proceedings of the 10th Innovations in Software Engineering Conference* (pp. 45-55). https://doi.org/10.1145/3021460.3021465

Silva, A., Araújo, T., Nunes, J., Perkusich, M., Dilorenzo, E., Almeida, H., & Perkusich, A. (2017). A systematic review on the use of definition of done on agile software development projects. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering* (pp. 364-373). ACM. https://doi.org/10.1145/3084226.3084262

Šmite, D., Moe, N., Floryan, M., Lavinta, G., & Chatzipetrou, P. (2020). Spotify guilds: When the value increases engagement, engagement increases the value. *Communications of the ACM*, *63*(3), 56-61. https://doi.org/10.1145/3343146

Šmite, D., Moe, N. B., Šāblis, A., & Wohlin, C. (2017). Software teams and their knowledge networks in large-scale software development. *Information and Software Technology*, *86*, 71-86.

Šmite, D., Moe, N. B., & Ågerfalk, P. J. (Eds.). (2010). *Agility across time and space: Implementing agile methods in global software projects*. Springer.

Smite, D., Wohlin, C., Gorschek, T., & Feldt, R. (2010). Empirical evidence in global software engineering: A systematic review. *Empirical Software Engineering, 15*(1), 91-118. https://doi.org/10.1007/s10664-009-9123-y

Stavru, S. (2014). A critical examination of recent industrial surveys on agile method usage. *Journal of Systems and Software, 94*, 87-97. https://doi.org/10.1016/j.jss.2014.03.041

Stojanov, I., Turetken, O., & Trienekens, J. J. (2015). A maturity model for scaling agile development. In *2015 41st Euromicro Conference on Software Engineering and Advanced Applications* (pp. 446-453). IEEE. https://doi.org/10.1109/seaa.2015.29

Stray, V. (2018). Planned and unplanned meetings in large-scale projects. In S. Wagner, R. Hoda, & A. Aguiar (Eds.), *Proceedings of the 19th International Conference on Agile Software Development*: *Companion* (Article 8). ACM. https://doi.org/10.1145/3234152.3234178

Stray, V. G., Moe N. B., & Dybå, T. (2012). Escalation of commitment: A longitudinal case study of daily meetings. In C. Wohlin (Ed.), *Lecture notes in business information processing: Vol 111. Agile processes in software engineering and extreme programming* (pp. 153-167). Springer. https://doi.org/10.1007/978-3-642-30350-0_11

Stray, V., Moe, N. B., & Hoda, R. (2018). Autonomous agile teams: Challenges and future directions for research. In S. Wagner, R. Hoda, & A. Aguiar (Eds.), *Proceedings of the 19th International Conference on Agile Software Development: Companion* (Article 16). https://doi.org/10.1145/3234152.3234182

Stray, V., Sjøberg, D. I., & Dybå, T. (2016). The daily stand-up meeting: A grounded theory study. *Journal of Systems and Software, 114*, 101-124. https://doi.org/10.1016/j.jss.2016.01.004

Strode, D. E., Huff, S. L., Hope, B., & Link, S. (2012). Coordination in co-located agile software development projects. *Journal of Systems and Software, 85*(6), 1222-1238. https://doi.org/10.1016/j.jss.2012.02.017

Sutherland, J. (2001). Agile can scale: Inventing and reinventing Scrum in five companies. *Cutter IT Journal, 14*(2), 5-11.

Sutherland, J., & Schwaber, K. (2011). *The Scrum papers: Nuts, bolts, and origins of an agile process*. http://scrumtraininginstitute.com

Sutherland, J., & Schwaber, K. (2013). *The Scrum guide: The definitive guide to Scrum; The rules of the game*. http://www.scrum.org

Swedish Research Council (2017). *Good research practice*. http://www.vr.se

Söderlund, J. (2005). Developing project competence: Empirical regularities in competitive project operations. *International Journal of Innovation Management, 9*(4), 451-480. https://doi.org/10.1142/s1363919605001344

Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard Business Review, 64*(1), 137-146.

Taxén, L., & Riedl, R. (2016). Understanding coordination in the information systems domain: Conceptualization and implications. *Journal of Information Technology Theory and Application, 7*(1), 5-40.

Tay, H. L. (2016). Lean improvement practices: Lessons from healthcare service delivery chains. *IFAC-PapersOnLine, 49*(12), 1158-1163. https://doi.org/10.1016/j.ifacol.2016.07.660

Thomas, G., & Myers, K. (2015). *The anatomy of the case study*. SAGE Publications.

Thompson, J. D. (1967). *Organizations in Action*. McGraw-Hill.

Thorén, C., Ågerfalk, P. J., & Edenius, M. (2014). Through the printing press: An account of open practices in the Swedish newspaper industry. *Journal of the Association for Information Systems, 15*(11), 779-804. https://doi.org/10.17705/1jais.00379

Thornton, P. H., & Ocasio, W. (2008). Institutional logics. In R. Greenwood, C. Oliver, K. Sahlin, & R. Suddaby, *The Sage handbook of organizational institutionalism* (pp. 99-128). SAGE publications.

Thornton, P. H., Ocasio, W., & Lounsbury, M. (2012). *The institutional logics perspective: A new approach to culture, structure, and process*. Oxford University Press.

Turk, D., France, R., & Rumpe, B. (2002/2014). Limitations of agile software processes. In *Third International Conference on Extreme Programming and Flexible Processes in Software Engineering, XP2002* (pp. 43-46). https://arxiv.org/ftp/arxiv/papers/1409/1409.6600.pdf

Uludağ, Ö., Kleehaus, M., Dreymann, N., Kabelin, C., & Matthes, F. (2019). Investigating the adoption and application of large-scale Scrum at a German automobile manufacturer. In *ACM/IEEE 14th International Conference on Global Software Engineering, ICGSE* (pp. 22-29). IEEE. https://doi.org/10.1109/icgse.2019.00019

Uludağ, Ö., Philipp, P., Putta, A., Paasivaara, M., Lassenius, C., & Matthes, F. (2020). Revealing the state-of-the-art in large-scale agile development: A systematic mapping Study. *Information and Software Technology*. https://arxiv.org/ftp/arxiv/papers/2007/2007.05578.pdf

Walsham, G. (2006). Doing interpretive research. *European Journal of Information Systems, 15*(3), 320-330. https://doi.org/10.1057/palgrave.ejis.3000589

Van Waardenburg, G., & Van Vliet, H. (2013). When agile meets the enterprise. *Information and software technology, 55*(12), 2154-2171. https://doi.org/10.1016/j.infsof.2013.07.012

Van de Ven, A. H., Delbecq, A. L., & Koenig Jr., R. (1976). Determinants of coordination modes within organizations. *American Sociological Review*, *41*(2), 322-338. https://doi.org/10.2307/2094477

Van der Blonk, H. (2003). Writing case studies in information systems research. *Journal of Information Technology, 18*(1), 45-52. https://doi.org/10.1080/0268396031000077440

VersionOne, 2020. *14th annual "State of Agile Development" survey*. http://www.versionone.com

Williams, L., & Cockburn, A. (2003). Agile software development: It's about feedback and change. *Computer, 36*(6), 39–43. https://doi.org/10.1109/mc.2003.1204373

Xu, P. (2009). Coordination in large agile projects. *The Review of Business Information Systems*, *13*(4), 29-43. https://doi.org/10.19030/rbis.v13i4.4310

Yin, R.K., (2017). *Case study research and applications: Design and methods* (6th ed.). SAGE Publications.

Ågerfalk, P. J., & Fitzgerald, B. (2006). Flexible and distributed software processes: Old petunias in new bowls? *Communications of the ACM, 49*(10), 27-34.

https://doi.org/10.1145/3262237

Ågerfalk, P. J., Fitzgerald, B., & Slaughter, S. A. (2009). Flexible and distributed information systems development: State of the art and research challenges. *Information Systems Research, 20*(3), 317–328. https://doi.org/10.1287/isre.1090.0244

Appendix A. Interview guide

1. About you:

- What is your role?
    - What does your team do?
- How long have you been working here?
- How long have you been working Agile?
- Have you received any education/training for Agile ways of working? What?

2. About large-scale Agile work and autonomy

- How do you experience the introduction of large-scale processes in your organization?
    - What has been difficult?
    - What has been simple (easier than you thought)?
    - Has the introduction of large-scale processes had the impact you expected?
    - Has it had any negative effects?
- How has autonomy in your/each team changed?
    - In what areas do you have more mandate?
    - In what areas do you have less mandate?
- What do you think you could do (change) to increase autonomy for each group but still manage coordination effectively (the problems with dependency between teams)? Speculate.

3. About dependencies and coordination

- Do you have much dependencies between teams?
- Do you have dependencies with other stakeholders/teams, outside your track?
- What type of dependencies are most common: sequential (ie someone else must make it clear before you can start) or competence (ie some experts have to do something because you cannot)
    - Estimate in %, how much of each dependency?
- Describe how different forms of coordination are performed:
    - scrum-of-scrums
        - Do you think they are being performed too often? Too seldom?
        - Experienced benefits?
        - Experienced difficulties?

- Something you would like to change with how they are performed?

o PI Planning occasions

- Do you think they are being performed too often? Too seldom?
- Experienced benefits?
- Experienced difficulties?
- Something you would like to change with how they are performed?

o Communities of practice?

- Do you think they are being performed too often? Too seldom?
- Experienced benefits?
- Experienced difficulties?
- Something you would like to change with how they are performed?

o Why have you chosen these forms of coordination?

- Experienced benefits?
- Experienced difficulties?
- Something you would like to change with how they are performed?

• Do you do anything further to coordinate (between teams)? Tell me.

o Other formal/structured/planned ways to coordinate?

o Other informal/ad-hoc ways to coordinate?

• What do you think helps most to facilitate coordination?

• What do you do to detect dependencies / co-ordination needs between planning opportunities?

o How is it documented and communicated? (Is your program board up-to-date?)

o How do you do with ”solved dependencies”, i.e. (how is it documented / communicated)?

• How many of the dependencies that exist between teams arise between planning occasions (ie surprises that you did not predict during the planning workshop)?

- Percentage - New vs. discovered at PI planning?
- Are there data on it? do you measure it?

4. About adaptations (tailoring)

- How do you think about adapting SAFe to your needs?
  - What is the ”minimum” to be SAFe in your opinion?
- What have you changed regarding processes, roles and artifacts compared to “standard” SAFe?

5. Management issues (to RTE:s and managers higher up in the hierarchy):

- How many teams do you have in your track?
- How have you put together the teams? (Feature? Component? Other?)
  - Has it changed along the way?
- How does product management work, ie:
  - Who is included?
  - How often do they meet?
  - Experienced benefits?
  - Experienced disadvantages?
- Why did you choose SAFe?
  - What benefits did you expect from SAFe?

Appendix B. Survey questionnaire

Questionnaire

What is your role? ... (e. g. Scrum master, PO, SW developer)

What is your age? ... Gender? ... Years worked at X?...

Name of your team? ... (or, ☐ I don't belong to a team)

Questions for <u>everyone</u>:

	Large-scale practices	Not true	Rarely true	Neu-tral	Usually true	Very true	*Don't know*
1	The PI planning gives me a good overview of our work	☐	☐	☐	☐	☐	☐
2	The program board is very helpful for coordination between teams	☐	☐	☐	☐	☐	☐
3	The Scrum of Scrum (SoS) meetings solve inter-team coordination problems	☐	☐	☐	☐	☐	☐
4	There are many problems with working in a large-scale setting as ours, using SAFe.	☐	☐	☐	☐	☐	☐
5	I would like to go back to the old way of working (instead of working according to SAFe).	☐	☐	☐	☐	☐	☐
6	What do you consider as the main **benefit** of working according to SAFe in your organization?						
7	What do you consider as the main **drawback** of working according to SAFe in your organization?						

Questions for <u>team members</u>:

	Now compared to before you started SAFe	Much worse	Worse	Same	Better	Much bette	*Don't know*
A	How does coordination *with other teams* work now, compared to one year ago (before implementing SAFe)?	☐	☐	☐	☐	☐	☐
B	How does communication *within the team* work now, compared to one year ago (before implementing SAFe)?	☐	☐	☐	☐	☐	☐
C	How is the *balance* between member contributions within the team now, compared to one year ago (before implementing SAFe)?	☐	☐	☐	☐	☐	☐
D	How does coordination *within the team* work now, compared to one year ago (before implementing SAFe)?	☐	☐	☐	☐	☐	☐
E	How does *support* within the team (helping each other) work now, compared to one year ago (before implementin SAFe)?	☐	☐	☐	☐	☐	☐
F	How is the cohesion within the team now, compared to one year ago (before implementing SAFe)?	☐	☐	☐	☐	☐	☐
G	How does the personal effort to achieve goals differ now, compared to one year ago (before implementing SAFe), within the team?	☐	☐	☐	☐	☐	☐
H	How does your team performance differ now, compared to one year ago (before implementing SAFe)?	☐	☐	☐	☐	☐	☐
I	In what way has trust and sense of personal security changed within the team, compared to one year ago (before implementing SAFe)?	☐	☐	☐	☐	☐	☐
	Inter-team coordination	Not true	Rarely true	Neu-tral	Usually true	Very true	*Don't know*
A1	Processes and activities are well coordinated with other teams	☐	☐	☐	☐	☐	☐
A2	Duplicated and overlapping activities are avoided	☐	☐	☐	☐	☐	☐
A3	Discussions with other teams are conducted constructively.	☐	☐	☐	☐	☐	☐
A4	Conflicts with other teams are settled quickly	☐	☐	☐	☐	☐	☐
	Communication within the team	Not true	Rarely true	Neu-tral	Usually true	Very true	*Don't know*
B1	We communicate often in spontaneous meetings	☐	☐	☐	☐	☐	☐
B2	Team members communicate mostly directly and personally with each other	☐	☐	☐	☐	☐	☐
B3	In our team, there are conflicts regarding the openness of the information flow.	☐	☐	☐	☐	☐	☐

B4	Relevant information is shared openly by all team members	☐	☐	☐	☐	☐	☐
	Contributions/Balance within the team	Not true	Rarely true	Neu-tral	Usually true	Very true	*Don't know*
C1	The team recognize the potentials (strengths and weaknesses) of individual team members	☐	☐	☐	☐	☐	☐
C2	Team members contribute to the goals in accordance with their specific potential	☐	☐	☐	☐	☐	☐
C3	Imbalance of member contributions cause conflicts in our team	☐	☐	☐	☐	☐	☐
	Coordination within the team During the last PI...	Not true	Rarely true	Neu-tral	Usually true	Very true	*Don't know*
D1	...there were clear and fully *understood* expectations for tasks within our team	☐	☐	☐	☐	☐	☐
D2	...expectations for tasks were *accepted* by team members	☐	☐	☐	☐	☐	☐
D3	...there were conflicting interests in our team regarding tasks	☐	☐	☐	☐	☐	☐
	Support within the team	Not true	Rarely true	Neu-tral	Usually true	Very true	*Don't know*
E1	Team members help and support each other as best they can	☐	☐	☐	☐	☐	☐
E2	If conflicts come up, they are easily and quickly resolved	☐	☐	☐	☐	☐	☐
E3	Discussions and controversies are conducted constructively	☐	☐	☐	☐	☐	☐
E4	Suggestions and contributions of team members are respected.	☐	☐	☐	☐	☐	☐
	Cohesion within the team	Not true	Rarely true	Neu-tral	Usually true	Very true	*Don't know*
F1	There are many personal conflicts in our team	☐	☐	☐	☐	☐	☐
F2	The members of our team feel proud to be part of the team	☐	☐	☐	☐	☐	☐
F3	Team members feel responsible for developing and protect-ing the team	☐	☐	☐	☐	☐	☐
	Effort	Not true	Rarely true	Neu-tral	Usually true	Very true	*Don't know*
G1	There were conflicts regarding the effort that team members put into the planned work for this PI	☐	☐	☐	☐	☐	☐
G2	Our team put much effort into reaching the team's objec-tives for this PI	☐	☐	☐	☐	☐	☐
G3	Every team member made the planned work for this PI their highest priority	☐	☐	☐	☐	☐	☐
	Team performance	Not true	Rarely true	Neu-tral	Usually true	Very true	*Don't know*
H1	Recently, this team seems to be "slipping" a bit in its level of performance.	☐	☐	☐	☐	☐	☐
H2	Those who receive or use the work this team does often have complaints about our work	☐	☐	☐	☐	☐	☐
H3	The quality of work provided by this team is improving over time	☐	☐	☐	☐	☐	☐
H4	Critical quality errors occur frequently in this team	☐	☐	☐	☐	☐	☐
	Team behaviours	Not true	Rarely true	Neu-tral	Usually true	Very true	*Don't know*
I1	If you make a mistake on this team, it is often held against you	☐	☐	☐	☐	☐	☐
I2	Members of this team are able to bring up problems and tough issues	☐	☐	☐	☐	☐	☐
I3	People on this team sometimes reject others for being different	☐	☐	☐	☐	☐	☐
I4	It is safe to take a risk on this team	☐	☐	☐	☐	☐	☐

Appendix C. Case study protocol

(Based on the outline of four sections proposed by Yin, 2017).

Section A. Overview of the Case Study

1. Mission and goal of the study
2. Case study questions, hypotheses, and propositions.
3. Theories important for the case study

Section B. Data Collection Procedures

1. Preparation prior to fieldwork
2. Data collection plan for interviews (roles of respondents)
3. Type of observations (attending PI planning event, attending Scrum of Scrums, other)

Section C. Protocol Questions

1. Implementation of inter-team coordination routines
    a. Which routines are implemented?
    b. How are they performed?
    c. Why are they implemented, rationale for implementing, desired effects?
2. Tailoring of inter-team coordination routines
    a. Which routines are tailored?
    b. How are they tailored?
    c. Why are they tailored, rationale for tailoring, desired effects?
3. Impacts of inter-team coordination routines
    a. What kind of benefits are perceived?
    b. What kind of drawbacks are perceived?

Section D. Tentative Outline for the Case Study Report

1. Chronology of events covering the implementation and tailoring of inter-team coordination routines
2. Lists of persons interviewed
3. Perceived impacts of implementing and tailoring inter-team coordination routines

Appendix D. SLR context descriptions

P#	Routines or framework	Industry	Organization	ASD organization size	Study type
P1	Scrum, SoS, CoP	Cloud Software Service	Salesforce	More than 30 teams	Experience report, Qualitative
P2	Scrum, SoS	Telecom	Ericsson Gothenburg	Not described	Case study, Qualitative
P3	LeSS	Telecom	Siemens in India and Germany	Not described	Observational study, Quantitative
P4	One of the cases: A mix of Nexus and LeSS	Financial services, SW vendor, Public organization	Not presented	Not described	Mixed method
P5	Not described	Telecom	Nokia Worldwide	More than 1000 people	Case study, Mixed method
P6	Not described	Telecom	Ericsson Linköping	420 + 120 people	Case study, Quantitative
P7	Not described	Broadcasting	BBC	9 teams	Qualitative
P8	Not described	Telecom, utility vehicles manufacturer, automotive	Not presented	More than 100 developers	Qualitative
P9	Not described	Software vendor for health services	Siemens HS	8 teams which grew to over 300 people	Qualitative
P10	Not described	Norwegian bank	Not presented	8 teams	Qualitative
P11	Not described	Telecom	Ericsson R&D Center Finland	350 people in two sites	Case study, Quantitative
P12	Scrum, SoS	Software vendor for oil and energy	Not presented	50 people in two sites	Qualitative
P13	LeSS	Telecom	Nokia Finland and Greece	20 teams	Case study, Qualitative
P14	Scrum, SoS	Telecom	Not presented	20 teams + 25 teams	Qualitative
P15	Scrum, SoS, CoP	Government agency	Not presented	120 people	Qualitative

P16	Scrum, SoS	Telecom	Ericsson AB	300 people	Qualitative
P17	Not described	Telecom	Ericsson AB	Several thousand developers	Qualitative
P18	Scrum, SoS	Government agency	Norwegian Pension Fund	12 teams (175 people)	Case study, Qualitative
P19	Scrum, SoS	Software vendor in healthcare	Not presented	120 people	Case study, Qualitative
P20	SAFe	Several	Finnish organizations	Not described	Quantitative
P21	SAFe	Software vendor	Comptel	Case1: 14 teams, Case2: 12 teams	Case studies, Qualitative
P22	Scrum, SoS	Telecom	Ericsson R&D	15 teams	Case study, Qualitative
P23	SAFe	Financial services	SimCorp	55 teams (500 employees)	Case study, Qualitative
P24	Spotify model	Software vendor	Not presented	6 teams	Case study, Qualitative
P25	Scrum, SoS, Lean values	Software vendor	SAP AG	18 000 software developers	Case study, Qualitative
P26	Spotify model	Cloud Software Service	Spotify	700 software developers	Case study, Mixed method
P27	LeSS	Automotive, Germany	Not presented	29 teams	Case study, Qualitative

The Swedish Research School of Management and Information Technology MIT

The *Swedish Research School of Management and Information Technology* (MIT) is one of 16 national research schools supported by the Swedish Government. MIT is jointly operated by the following institutions: Blekinge Institute of Technology, Chalmers University of Technology, University of Gothenburg, Jönköping International Business School, Karlstad University, Linköping University, Linnaeus University Växjö, Lund University, Mälardalen University College, Stockholm University, Umeå University, Örebro University, and Uppsala University, host to the research school. At the Swedish Research School of Management and Information Technology (MIT), research is conducted, and doctoral education provided, in three fields: management information systems, business administration, and informatics.

DISSERTATIONS FROM THE SWEDISH RESEARCH SCHOOL OF MANAGEMENT AND INFORMATION TECHNOLOGY

Doctoral theses (2003-)

1. **Baraldi, Enrico (2003),** *When Information Technology Faces Resource Interaction: Using IT Tools to Handle Products at IKEA and Edsbyn.* Department of Business Studies, Uppsala University, Doctoral Thesis No. 105.

2. **Wang, Zhiping (2004),** *Capacity-Constrained Production-Inventory Systems: Modelling and Analysis in both a Traditional and an E-Business Context*. IDA-EIS, Linköpings universitet och Tekniska Högskolan i Linköping, Dissertation No. 889

3. **Ekman, Peter (2006),** *Enterprise Systems & Business Relationships: The Utilization of IT in the Business with Customers and Suppliers*. School of Business, Mälardalen University, Doctoral Dissertation No 29.

4. **Lindh, Cecilia (2006),** *Business Relationships and Integration of Information Technology*. School of Business, Mälardalen University, Doctoral Dissertation No 28.

5. **Frimanson, Lars (2006),** *Management Accounting and Business Relationships from a Supplier Perspective.* Department of Business Studies, Uppsala University, Doctoral Thesis No. 119.

6. **Johansson, Niklas (2007),** *Self-Service Recovery*. Information Systems, Faculty of Economic Sciences, Communication and IT, Karlstad University, Dissertation KUS 2006:68.

7. **Sonesson, Olle (2007),** *Tjänsteutveckling med personal medverkan: En studie av banktjänster.* Företagsekonomi, Fakulteten för ekonomi, kommunikation och IT, Karlstads universitet, Doktorsavhandling, Karlstad University Studies 2007:9.

8. **Maaninen-Olsson, Eva (2007),** *Projekt i tid och rum: Kunskapsintegrering mellan projektet och dess historiska och organisatoriska kontext.* Företagsekonomiska institutionen, Uppsala universitet, Doctoral Thesis No. 126.

9. **Keller, Christina (2007),** *Virtual learning environments in higher education: A study of user acceptance.* Linköping Studies in Science and Technology, Dissertation No. 1114.

10. **Abelli, Björn (2007),** *On Stage! Playwriting, Directing and Enacting the Informing Processes*. School of Business, Mälardalen University, Doctoral Dissertation No. 46.

11. **Cöster, Mathias (2007),** *The Digital Transformation of the Swedish Graphic Industry*. Linköping Studies in Science and Technology, Linköping University, Dissertation No. 1126.

12. **Dahlin, Peter (2007),** *Turbulence in Business Networks: A Longitudinal Study of Mergers, Acquisitions and Bankruptcies Involving Swedish IT-companies*. School of Business, Mälardalen University, Doctoral Thesis No. 53.

13. **Myreteg, Gunilla (2007),** *Förändringens vindar: En studie om aktörsgrupper och konsten att välja och införa ett affärssystem*. Företagsekonomiska institutionen, Uppsala universitet, Doctoral Thesis No. 131.

14. **Hrastinski, Stefan (2007),** *Participating in Synchronous Online Education*. School of Economics and Management, Lund University, Lund Studies in Informatics No. 6.

15. **Granebring, Annika (2007),** *Service-Oriented Architecture: An Innovation Process Perspective.* School of Business, Mälardalen University, Doctoral Thesis No. 51.

16. **Lövstål, Eva (2008),** *Management Control Systems in Entrepreneurial Organizations: A Balancing Challenge*. Jönköping International Business School, Jönköping University, JIBS Dissertation Series No. 045.

17. **Hansson, Magnus (2008),** *On Closedowns: Towards a Pattern of Explanation to the Closedown Effect*. Örebro University School of Business, Örebro University, Doctoral Thesis No. 1.

18. **Fridriksson, Helgi-Valur (2008),** *Learning processes in an inter-organizational context: A study of krAft project*. Jönköping International Business School, Jönköping University, JIBS Dissertation Series No. 046.

19. **Selander, Lisen (2008),** Call Me *Call Me for some Overtime: On Organizational Consequences of System Changes.* Institute of Economic Research, Lund Studies in Economics and Management No. 99.

20. **Henningsson, Stefan (2008),** *Managing Information Systems Integration in Corporate Mergers & Acquisitions*. Institute of Economic Research, Lund Studies in Economics and Management No. 101.

21. **Ahlström, Petter (2008),** *Strategier och styrsystem för seniorboende-marknaden*. IEI-EIS, Linköping universitetet och Tekniska Högskolan i Linköping, Doktorsavhandling, Nr. 1188.

22. **Sörhammar, David (2008),** *Consumer-firm business relationship and network: The case of "Store" versus Internet.* Department of Business Studies, Uppsala University, Doctoral Thesis No. 137.

23. **Caesarius, Leon Michael (2008),** *In Search of Known Unknowns: An Empirical Investigation of the Peripety of a Knowledge Management System*. Department of Business Studies, Uppsala University, Doctoral Thesis No. 139.

24. **Cederström, Carl (2009),** *The Other Side of Technology: Lacan and the Desire for the Purity of Non-Being.* Institute of Economic Research, Lund University, Doctoral Thesis, ISBN: 91-85113-37-9.

25. **Fryk, Pontus, (2009),** *Modern Perspectives on the Digital Economy: With Insights from the Health Care Sector*. Department of Business Studies, Uppsala University, Doctoral Thesis No. 145.

26. **Wingkvist, Anna (2009)**, *Understanding Scalability and Sustainability in Mobile Learning: A Systems Development Framework.* School of Mathematics and Systems Engineering, Växjö University, Acta Wexionesia, No. 192, ISBN: 978-91-7636-687-5.

27. **Sällberg, Henrik (2010***), Customer Rewards Programs: Designing Incentives for Repeated Purchase.* Blekinge Institute of Technology, School of Management, Doctoral Dissertation Series No. 2010:01.

28. **Verma, Sanjay (2010),** *New Product Newness and Benefits: A Study of Software Products from the Firms' Perspective*. Mälardalen University Press, Doctoral Thesis.

29. **Iveroth, Einar (2010),** *Leading IT-Enabled Change Inside Ericsson: A Transformation Into a Global Network of Shared Service Centres*. Department of Business Studies, Uppsala University, Doctoral Thesis No. 146.

30. **Nilsson, Erik (2010),** *Strategi, styrning och konkurrenskraft: En longitudinell studie av Saab AB*. IEI-EIS, Linköpings universitet och Tekniska Högskolan i Linköping, Doktorsavhandling, Nr. 1318.

31. **Sjöström, Jonas (2010),** *Designing Information Systems: A pragmatic account*. Department of Informatics and Media, Uppsala University, Doctoral Thesis.

32. **Numminen, Emil (2010),** *On the Economic Return of a Software Investment: Managing Cost, Benefit and Uncertainty*. Blekinge Institute of Technology, School of Management, Doctoral Thesis.

33. **Frisk, Elisabeth (2011),** *Evaluating as Designing: Towards a Balanced IT Investment Approach*. IT University, Göteborg, Doctoral Thesis.

34. **Karlsudd, Peter (2011),** *Support for Learning: Possibilities and Obstacles in Learning Applications*. Mälardalen University, Doctoral Thesis.

35. **Wicander, Gudrun (2011),** *Mobile Supported e-Government Systems: Analysis of the Education Management Information System (EMIS) in Tanzania*. Karlstad University, Doctoral Thesis. Karlstad University Studies 2011:49.

36. **Åkesson, Maria (2011),** *Role Constellations in Value Co-Creation: A Study of Resource Integration in an e-Government Context*. Karlstad University, Doctoral Thesis. Karlstad University Studies 2011:36.

37. **Nfuka, Edephonce N. (2012),** *IT Governance in Tanzanian Public Sector Organisations*. Department of Computer and Systems Sciences, Stockholm University**,** Doctoral Thesis.

38. **Larsson, Anders Olof (2012),** *Doing Things in Relation to Machines: Studies on Online Interactivity.* Department of Informatics and Media, Uppsala University, Doctoral Thesis.

39. **Andersson, Bo (2012),** *Harnessing Handheld Computing: Framework, Toolkit and Design Propositions.* Lund University, Doctoral Thesis**.**

40. **Erixon, Cecilia (2012),** *Information System Providers and Business Relationships: A Study on the Impact of Connections*. Mälardalen University, Doctoral Thesis.

41. **Svensson, Martin (2012)**, *Routes, Routines and Emotions in Decision Making of Emergency Call Takers*. Blekinge Institute of Technology, Doctoral Dissertation Series No. 2012:04.

42. **Svensson, Ann (2012),** *Kunskapsintegrering med informationssystem I professionsorienterade praktiker*. Institutionen för tillämpad IT, Göteborgs universitet, Doktorsavhandling.

43. **Pareigis, Jörg (2012),** *Customer Experiences of Resource Integration: Reframing Servicescapes Using Scripts and Practices*. Karlstad University, Doctoral Thesis. Karlstad University Studies 2012:38.

44. **Röndell, Jimmie (2012),** *From Marketing to, to Marketing with Consumers*. Department of Business Studies, Uppsala University, Doctoral Thesis No. 155.

45. **Lippert, Marcus (2013),** *Communities in the Digital Age: Towards a Theoretical Model of Communities of Practice and Information Technology*. Department of Business Studies, Uppsala University, Doctoral Thesis No. 156.

46. **Netz, Joakim (2013),** *Diffusa spänningar eller spännande tillväxt? Företagsledning i tider av snabb förändring*. Mälardalens högskola, Doktorsavhandling nr 135.

47. **Thorén, Claes (2013),** *Print or Perish? A Study of Inertia in a Regional Newspaper Industry*. Karlstad University, Doctoral Thesis. Karlstad University Studies 2014:10 (Ny uppl.).

    Stockhult, Helén (2013), *Medarbetare i dialog: en studie om viljan att göra mer än det formellt förväntade*. Örebro universitet, Örebro Studies in Business Dissertations, 4.

48. **Mihailescu, Daniela (2013),** *Explaining the Use of Implementation Methodology in Enterprise Systems Implementation Context: A Critical Realist Perspective*. Lund University, Doctoral Thesis.

49. **Ghazawneh, Ahmad (2012),** *Towards a Boundary Resources Theory of Software Platforms*. Jönköping International Business School, Doctoral Thesis.

50. **Shams, Poja (2013),** *What Does it Take to Get your Attention? The Influence of In-Store and Out-of-Store Factors on Visual Attention and Decision Making for Fast-Moving Consumer Goods*. Karlstad University, Doctoral Thesis. Karlstad University Studies 2013:5.

51. **Osowski, Dariusz (2013),** *From Illusiveness to Genuineness: Routines, Trading Zones, Tools and Emotions in Sales Work*. Department of Business Studies, Uppsala University, Doctoral Thesis No. 160.

52. **Höglund, Linda (2013)**, *Discursive Practises in Strategic Entrepreneurship: Discourses and Repertoires in Two Firms*. Örebro University, Doctoral Thesis.

53. **Persson Ridell, Oscar (2013)**, *Who is the Active Consumer? Insight into Contemporary Innovation and Marketing Practices*. Department of Business Studies, Uppsala University, Doctoral Thesis.

54. **Kask, Johan (2013),** *On business relationships as Darwinian systems: An exploration into how Darwinian systems thinking can support business relationship research*. Örebro University, Doctoral Thesis.

55. **Paulsson, Wipawee Victoria (2013)**, *The Complementary Use of IS Technologies to Support Flexibility and Integration Needs in Budgeting*. Lund University, Doctoral Thesis.

56. **Kajtazi, Miranda (2013),** *Assessing Escalation of Commitment as an Antecedent of Noncompliance with Information Security Policy*. Linnaeus University, Doctoral Thesis.

57. **Hasche, Nina (2013),** *Value Co-Creating Processes in International Business Relationships: Three empirical stories of co-operation between Chinese customers and Swedish suppliers*. Örebro University, Doctoral Thesis.

58. **Pierce, Paul (2013),** *Using Alliances to Increase ICT Capabilities*. Lund University, Doctoral Thesis.

59. **Mansour, Osama (2013),** *The Bureaucracy of Social Media: An Empirical Account in Organizations*. Linnaeus University, Doctoral Thesis.

60. **Osmonalieva, Zarina (2013)**, *Factors Determining Exploitation of Innovative Venture Ideas: A study of nascent entrepreneurs in an advisory system*. Mälardalen University, Doctoral Thesis.

61. **Holmberg, Nicklas (2014),** *The Purity of Separation of Concerns: The Service Oriented Business Process - a Design Approach for Business Agility.* Lund University, Doctoral Thesis.

62. **Poth, Susanna (2014),** *Competitive Advantage in the Service Industry. The Importance of Strategic Congruence, Integrated Control and Coherent Organisational Structure: A Longitudinal Case Study of an Insurance Company*. Department of Business Studies, Uppsala University, Doctoral Thesis.

63. **Safari, Aswo (2014),** *Consumer Foreign Online Purchasing: Uncertainty in the Consumer-Retailer Relationship*. Department of Business Studies, Uppsala University, Doctoral Thesis.

64. **Sandberg, Johan (2014)**, *Digital Capability: Investigating Coevolution of IT and Business Strategies*. Umeå University, Doctoral Thesis.

65. **Eklinder Frick, Jens (2014),** *Sowing Seeds for Innovation: The Impact of Social Capital in Regional Strategic Networks*. Mälardalen University, Doctoral Thesis.

66. **Löfberg, Nina (2014)**, *Service Orientation in Manufacturing Firms: Understanding Challenges with Service Business Logic*. Karlstad University, Doctoral Thesis. Karlstad University Studies 2014:30.

67. **Gullberg, Cecilia (2014)**, *Roles of Accounting Information in Managerial Work*. Department of Business Studies, Uppsala University, Doctoral Thesis No. 171.

68. **Bergkvist, Linda (2014)**, *Towards a Framework for Relational-Oriented Management of Information Systems Outsourcing: Key Conditions Connected to Actors, Relationships and Process*. Karlstad University, Doctoral Thesis. Karlstad University Studies 2014:31.

69. **Tavassoli, Sam (2014)**, *Determinants and Effects of Innovation: Context Matters.* Blekinge Institute of Technology, Doctoral Thesis No. 2014:10.

70. **Högström, Claes (2014)**, *Fit In to Stand Out: An Experience Perspective on Value Creation*. Karlstad University, Doctoral Thesis. Karlstad University Studies 2014:44.

71. **Jansson, Tomas (2015),** *Agila projektledningsmetoder och motivation*. Karlstads universitet, Doctoral Thesis. Karlstad University Studies 2015:9.

72. **Ryzhkova, Natalia (2015)**, *Web-Enabled Customer Involvement: A Firms' Perspective.* Blekinge Institute of Technology, Doctoral Thesis.

73. **Sundberg, Klas (2015),** *Strategisk utveckling och ekonomistyrning: Ett livscykelperspektiv*. Företagsekonomiska institutionen, Uppsala universitet, Doctoral Thesis No. 173.

74. **Nylén, Daniel (2015),** *Digital Innovation and Changing Identities: Investigating Organizational Implications of Digitalization*. Umeå University, Doctoral Thesis.

75. **Chowdhury, Soumitra (2015)**, *Service Logic in Digitalized Product Platforms: A Study of Digital Service Innovation in the Vehicle Industry*. Gothenburg University, Doctoral Thesis.

76. **Jogmark, Marina (2015),** *Den regionala transformationsprocessens sociala dimension. Karlskrona 1989-2002*. Blekinge Tekniska Högskola, Doctoral Thesis.

77. **Sundström, Angelina (2015),** *Old Swedish Business in New International Clothes: Case Studies on the Management of Strategic Resources in Foreign-Acquired Swedish R&D Firms*. Mälardalen University, Doctoral Thesis.

78. **Öbrand, Lars (2015),** *Information Infrastructure Risk: Perspectives, Practices & Technologies*. Umeå University, Doctoral Thesis.

79. **Brozović, Danilo (2016),** *Service Provider Flexibility: A Strategic Perspective*. Stockholm University, Doctoral Thesis.

80. **Siegert, Steffi (2016),** *Enacting Boundaries through Social Technologies: A Dance between Work and Private Life*. Stockholm University, Doctoral Thesis.

81. **Linton, Gabriel (2016)**, *Entrepreneurial Orientation: Reflections from a Contingency Perspective*. Örebro University, Doctoral Thesis.

82. **Akram, Asif (2016)**, *Value Network Transformation: Digital Service Innovation in the Vehicle Industry*. Department of Applied Information Technology, Chalmers University of Technology and University of Gothenburg, Doctoral Thesis.

83. **Hadjikhani, Annoch (2016)**, *Executive Expectation in the Internationalization Process of Banks: The Study of Two Swedish Banks Foreign Activities*. Department of Business Studies, Uppsala University, Doctoral Thesis No. 177.

84. **El-Mekawy, Mohamed (2016)**, *From Theory to Practice of Business-IT Alignment: Barriers, an Evaluation Framework and Relationships with Organizational Culture*. DSV, Stockholm University, Doctoral Thesis.

85. **Salavati, Sadaf (2016)**, *Use of Digital Technologies in Education: The Complexity of Teachers' Everyday Practice*. Linnaeus University, Doctoral Thesis.

86. **Pashkevich, Natallia (2016)**, *Information Worker Productivity Enabled by IT System Usage: A Complementary-Based Approach*. Stockholm Business School, Stockholm University, Doctoral Thesis.

87. **Stone, Trudy-Ann (2016)**, *Firms in Global Value Chains.* Blekinge Institute of Technology (BTH), Doctoral Thesis.

88. **Saarikko, Ted (2016)**, *An Inquiry into the Nature and Causes of Digital Platforms.* Umeå University, Doctoral Thesis.

89. **Tona, Olgerta (2017)**, *The Journey of Mobile Business Intelligence: From Vision to Use*. Lund University, Doctoral Thesis.

90. **Fredin, Sabrina (2017)**, *History and Geography Matter: The Cultural Dimension of Entrepreneurship*. Blekinge Institute of Technology, Doctoral Thesis.

91. **Giovacchini, Elia (2017)**, *Weaving the Symbiotic Relationship: A Longitudinal Study of a Firm-Sponsored Open Source Community Relationship Maintenance*. Stockholm Business School, Stockholm University, Doctoral Thesis.

92. **Gillmore, Edward (2017)**, *Four Essays on Subsidiary Evolution: Exploring the Antecedents, Contexts and Outcomes of Mandate Loss*. School of Business, Mälardalen University, Doctoral Thesis.

93. **Crawford, Jason (2017),** *Regulation's Influence on Risk Management and Management Control Systems in Banks*. Department of Business Studies, Uppsala University, Doctoral Thesis.

94. V**on Schantz, Hanna (2017),** *Well, that makes sense! Investigating opportunity development in a technology start-up.* Stockholm Business School, Stockholm University, Doctoral Thesis.

95. **Wass, Sofie (2017)**, *The Importance of eHealth Innovations: Lessons about Patient Accessible Information*. Jönköping International Business School, Doctoral Thesis.

96. **Imre, Özgün (2018)**, *Adopting Information Systems: Perspectives from Small Organizations*. Department of Management and Engineering (IEI), Linköping University, Doctoral Thesis.

97. **Lövgren, Daniel (2017)**, *Dancing Together Alone: Inconsistencies and Contradictions of Strategic Communication in Swedish Universities*. Informatics and Media, Uppsala University, Doctoral Thesis.

98. **Charitsis, Vasileios (2018)**, *Self-Tracking, Datafication and the Biopolitical Prosumption of Life.* Karlstad University, Doctoral Thesis.

99. **Lammi, Inti (2018)**, *A Practice Theory in Practice: Analytical Consequences in the Study of Organization and Socio-Technical Change.* Department of Business Studies, Uppsala University, Doctoral Thesis.

100. **Leite, Emilene (2018)**, *Complexity in the 'Extended' Business Network: A study of Business, Social and Political Relationships in Smart City Solutions*. Department of Business Studies, Uppsala University, Doctoral Thesis.

101. **Aasi, Parisa (2018)**, *Information Technology Governance: The Role of Organizational Culture and Structure*. Department of Computer and Systems Sciences, Stockholm University, Doctoral Thesis.

102. **Servadio, Luigi (2018)**, *Customer Rituals: Ethnographic Explorations of Wine Rituals with Families and Friends.* Stockholm Business School, Stockholm University, Doctoral Thesis.

103. **Ahlgren, Kajsa (2018)**, *Travelling Business Models: On Adapting Business Models to New Contexts.* Design Sciences, Faculty of Engineering, Lund University, Doctoral Thesis.

104. **Markowski, Peter (2018)**, *Collaboration Routines: A Study of Interdisciplinary Healthcare*. Stockholm Business School, Stockholm University, Doctoral Thesis.

105. **Zaffar, Fahd Omair (2018)**, *The Value of Social Media: What Social Networking Sites Afford Organizations*. Division of Informatics, Department of Applied Information Technology, University of Gothenburg, Doctoral Thesis.

106. **Stendahl, Emma (2018)**, *Headquarters Involvement in Managing Subsidiaries.* Stockholm Business School, Stockholm University, Doctoral Thesis.

107. **Fischer, Christian (2018)**, *Business Intelligence through a Sociomaterial Lens: The Imbrication of People and Technology in a Sales Process*. Department of Business Studies, Uppsala University, Doctoral Thesis.

108. **Lagin, Madelen (2018)**, *The Price We Pay: The Autonomy of Store Managers in Making Price Decisions*. Department of Business Studies, Örebro University, Doctoral Thesis.

109. **Odar, Susanne (2019)**, *Managementinitiativ, mening och verksamhetsresultat: En retrospektiv studie av en teknikintensiv verksamhet*. Department of Management and Engineering (IEI), Linköping University, Linköping Studies in Science and Technology, Doctoral Thesis.

110. **Radits, Markus (2019)**, *A Business Ecology Perspective on Community-Driven Open Source: The Case of the Free and Open Source Content Management System Joomla*. Department of Management and Engineering (IEI), Linköping University, Linköping Studies in Science and Technology, Doctoral Thesis No. 1937.

111. **Skog, Daniel A. (2019)**, *The Dynamics of Digital Transformation: The Role of Digital Innovation, Ecosystems and Logics in Fundamental Organizational Change*. Umeå University, Doctoral Thesis.

112. **Ek, Peter (2019)**, *Managing Digital Open Innovation with User Communities: A Study of Community Sensing and Product Openness Capabilities in the Video Game Industry*. Department of Business Studies, Uppsala University, Doctoral Thesis No. 199

113. **Muhic, Mirella (2019)**, *Transition to Cloud sourcing – Innovation and competitive advantage*. Design Sciences, Faculty of Engineering, Lund University, Doctoral Thesis.

114. **Mankevich, Vasili (2019)**, *Digital Innovation Management: Investigating Digital Trace Data in Online Communities*. Umeå University, Doctoral Thesis.

115. **Vink, Josina (2019)**, *In/visible - Conceptualizing Service Ecosystem Design*. Karlstad University, Doctoral Thesis.

116. **Bäckström, Izabelle (2019)**, *Mirror, mirror on the wall, who's the innovator after all? An explorative study of a management-initiated employee innovation process*. Department of Design Sciences, Faculty of Engineering, Lund University, Doctoral Thesis No. 116.

117. **Bani-Hani, Imad (2020)**, *Self-Service Business Analytics and the Path to Insights: Integrating Resources for Generating Insights,* Department of Informatics, School of Economics and Management, Lund University, Doctoral Thesis.

118. **Kashyap, Shruti Rangan (2020)**, *Monsoon Paper Dragons: Transparency, Accountability, Risk, and Compliance in Banking Regulation and Practice*, Department of Business Studies, Uppsala University, Doctoral Thesis No. 201.

119. **Havemo, Emelie (2020)**, *Den visuella bilden av organisationen: Perspektiv på visualitet i accounting*, Linköping Studies in Science and Technology, Institutionen för ekonomisk och industriell utveckling, Linköpings universitet, Doktorsavhandling nr. 2075.

120. **Kurti, Erdelina (2020)**, *Institutional Tensions and Complexity in the Digital Innovation of Incumbent Business Models*, Linnaeus University, Doctoral Thesis.

121. **Gustavsson, Tomas (2020)**, *Inter-team Coordination in Large-Scale Agile Software Development Projects*, Karlstad University, Doctoral Thesis.

Inter-team Coordination in Large-Scale Agile Software Development Projects

Software development organizations are implementing Agile values, principles, and frameworks all over the world. One important part of Agile is to allow teams to self-organize with a high level of autonomy; a strategy that has proved to be very successful. The Agile methods were initially designed for use in small, single-team projects. Routines for coordination between teams have not been adopted as much as routines for coordination within teams and have therefore not been much studied. A problem with several teams coordinating is that the autonomy must, to some extent, be sacrificed in the individual team when work needs to be coordinated with other teams, and a project is often part of a portfolio or program.

This thesis provides insights and rich descriptions of routines for inter-team coordination in large-scale Agile software development projects. How they are performed, how and why they are tailored, and the impacts of these added routines.

Empirical case studies have been performed at three large and diverse organizations: A company in the automotive industry, a business bank, and a Swedish government agency. Insights from this multiple-case study build on coordination theory as well as institutional logics (new institutional theory).

ISBN 978-91-7867-151-9 (print)

ISBN 978-91-7867-155-7 (pdf)

ISSN 1403-8099

DOCTORAL THESIS | Karlstad University Studies | 2020:30