

2022

Changes to team autonomy in large-scale software development: a multiple case study of Scaled Agile Framework (SAFe) implementations

Tomas Gustavsson
Karlstad University

Marthe Berntzen
University of Oslo

Viktoria Stray
University of Oslo, SINTEF

Follow this and additional works at: <https://aisel.aisnet.org/ijispm>

Recommended Citation

Gustavsson, Tomas; Berntzen, Marthe; and Stray, Viktoria (2022) "Changes to team autonomy in large-scale software development: a multiple case study of Scaled Agile Framework (SAFe) implementations," *International Journal of Information Systems and Project Management*. Vol. 10: No. 1, Article 3. Available at: <https://aisel.aisnet.org/ijispm/vol10/iss1/3>

This material is brought to you by AIS Electronic Library (AISeL). It has been accepted for inclusion in International Journal of Information Systems and Project Management by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.



Changes to team autonomy in large-scale software development: a multiple case study of Scaled Agile Framework (SAFe) implementations

Tomas Gustavsson

Karlstad University
Universitetsgatan 2, 651 88 Karlstad, Sweden
tomas.gustavsson@kau.se

Marthe Berntzen

University of Oslo
Gaustadalléen 23 B, 0373 Oslo, Norway
marthenb@ifi.uio.no

Viktoria Stray

University of Oslo, SINTEF
Gaustadalléen 23 B, 0373 Oslo, Norway
stray@ifi.uio.no

Abstract:

Large-scale transformations of agile ways of working have received more attention in the industry in recent years. Some organizations have developed their own solutions for scaling, whereas many have chosen trademarked frameworks. In large-scale agile software development, many developers and development teams carry out work simultaneously. When autonomous teams need to coordinate toward a common goal, they must sacrifice some level of autonomy. Development, testing, and integrations need to be coordinated with other teams and aligned with an organization's programs or portfolio. Through the conducting of 28 interviews and 17 on-site visits, this multiple case study explored how team autonomy changed in three agile software development organizations that implemented the Scaled Agile Framework (SAFe). The positive changes to team autonomy that they experienced as a result included getting a better overview, making better long-term decisions, giving and receiving help, and signaling limitations. We found two negative impacts on team autonomy: limited feature choice and enforced refinement. The study extends previous research on large-scale agile software development and improves our understanding of impacts on team autonomy.

Keywords:

agile software development; team autonomy; Scaled Agile Framework; SAFe; large-scale agile.

DOI: 10.12821/ijispm100102

Manuscript received: 23 July 2021

Manuscript accepted: 21 February 2022

1. Introduction

Since agile was introduced more than 20 years ago, it has become highly popular in not only small organizations but also larger organizations [1]. Today, the term “large-scale agile” is used to refer to agile practices and methods used beyond a single development team both at the inter-team and the organizational level [1]. In large-scale agile software development (ASD) settings, where several teams cooperate toward a common goal, new problems arise, such as dealing with dependencies between activities and ensuring shared goals between teams [1]. Typically, a software development team needs to coordinate with other teams regarding the constraints of, for example, requirements, testing, and integration [2]. These dependencies may limit the amount of autonomy and empowerment of the individual team. Therefore, the need to coordinate work processes with the rest of the organization reduces team autonomy [3]. A project is often part of a larger set of development projects, which requires feature delivery to be coordinated across teams. Thus, when self-organizing teams work together in such a setting, some level of autonomy needs to be sacrificed [4].

The balance between the benefits of autonomous, empowered teams and coordinated efforts toward a common goal is an important problem to solve for today’s software industry [5]. Practices for inter-team coordination have been proposed in the agile community to reduce the negative impacts of agile while also maintaining the positive impacts of teamwork stemming from agile ways of working [5]. These practices include, for example, scheduled and unscheduled meetings, as well as the use of shared visualization tools [6]. Moreover, many organizations use large-scale agile frameworks that come with pre-determined coordination practices and tools aimed at improving the inter-team coordination of software development. Many such frameworks are available, and among these, the most commonly adopted framework for large-scale agile today is, by far, the Scaled Agile Framework (SAFe) [7].

SAFe, however, has been criticized for the risk of delimiting autonomy for a single team [8, 9]. Limiting team autonomy could negatively impact teamwork and the team's performance, but these impacts have not been investigated much in large-scale ASD [10]. Based on the importance of team autonomy in large-scale ASD, the purpose of this study was to explore the impacts of scaling up agile ways of working on team autonomy. The research question of this study was formulated as follows:

How does team autonomy change when a large-scale agile framework is implemented?

The remainder of the paper is organized as follows: In Section 2, background information on autonomous agile teams and large-scale agile is presented. In Section 3, the research method is described. Section 4 presents results from the multiple-case study of three large-scale agile software development implementations. These findings are discussed in Section 5. Finally, Section 6 concludes the paper and presents key findings from the study.

2. Agile software development and team autonomy

2.1. Team autonomy in large-scale agile

The term “agile” refers to iterative and incremental practices stemming from a set of values and practices guiding modern approaches to software development. At the core of agile lies the notion of a self-managing team that has the autonomy to decide which practices are used and when to develop a software product [11]. Autonomous teams are defined as teams that have authority over their own work and have responsibility for aspects such as task assignment, planning, and scheduling [12]. In other words, these are teams that have the autonomy to decide which tasks to work on, when, and how.

Studies on team autonomy present several benefits for team members. Research has shown that highly autonomous teams are also more productive, are more creative, and have higher levels of team satisfaction and team commitment [13, 14]. A high level of team autonomy has also been related to higher levels of employee motivation and job satisfaction [15], as well as lower levels of stress [16]. According to the notion of team autonomy in the agile manifesto, self-organizing teams are the source of “the best architectures, requirements, and designs” [17].

However, scaling up agile software development also requires increased coordination needs. A larger software product often has a more complex architecture and code base, and many teams contribute to developing the overall product. This introduces dependencies, which need to be managed [18].

2.2. The Scaled Agile Framework (SAFe)

Many organizations use large-scale development frameworks to manage inter-team development challenges while at the same time ensuring some level of agility in the development process. Among these, SAFe is a popular commercial framework [7]. SAFe is intended to guide enterprises in scaling lean and agile practices by suggesting principles, organizational structure, and workflow patterns promoting alignment and collaboration. The framework describes agile ways of working on different levels, starting from the team level to the program level, portfolio level, and organizational level [19]. In addition, more roles are suggested on different levels. Two of these additional roles on the program level are the system architect, who coordinates architectural decisions, and the *release train engineer* (RTE), who coordinates a cluster of teams working together, known as the *agile release train* [19].

SAFe provides a range of inter-team coordination practices within an agile release train [19]. These inter-team practices include product increment (PI) planning meetings, Scrum of Scrums, and the use of a program board. The PI planning practice consists of a two-day workshop involving all teams and stakeholders who produce a release plan spanning eight to twelve weeks. The Scrum of Scrums practice is a face-to-face coordination meeting between representatives from each team performed several times per week. Finally, the program board visualizes dependencies between teams, both within the agile release train and external dependencies. The program board is produced during PI planning and is often updated in Scrum of Scrums meetings [19].

Recent reviews show that among studies on large-scale agile, SAFe has received the most research attention compared with other large-scale frameworks or studies of large-scale hybrid practices [10, 20, 21].

2.3. Threats to team autonomy in large-scale agile

When several agile teams cooperate toward a common goal, problems arise, such as dealing with dependencies between activities and ensuring shared goals between teams [18]. For example, an ASD team needs to coordinate with other ASD teams when it comes to the constraints of requirements, testing, and integration [2], and it cannot have full authority over all aspects of the work as a single team can [22]. However, coordination across teams has been identified as a major obstacle to large-scale agile's success [1, 20]. Coordination can be defined as the management of dependencies [23] achieved through the use of various mechanisms and practices aimed at resolving these dependencies. Coordination has received much research attention in the agile community [4, 21]. For example, a theory of coordination in co-located ASD teams has been developed, proposing that agile development tools and practices, such as stand-up meetings, backlogs, and team wallboards, contribute to coordination effectiveness within ASD teams [24]. However, in large-scale agile, the need for coordination between teams introduces more and different coordination mechanisms, such as the use of inter-team meetings and roles to ensure a shared overview across teams [25].

Without inter-team coordination, teams might be autonomous but ineffective [26]. For example, Moe et al. [22] describe the case of an ASD team that shielded itself by participating only in activities that it considered to be necessary for the team to take part in. If this prevents a single ASD team from being coordinated with overall plans and goals, it will affect other teams and deliverables. Still, introducing shared goals, activities, and processes is a threat to team autonomy, which is the key to agile ways of working [22].

Along with other teams, decisions from other roles affect a single ASD team. When a large-scale agile framework is implemented, new roles are often introduced, such as those of product managers or architects [27]. The product manager prioritizes the requirements that the teams and architects must develop to clarify the indented technical direction [27]. Dingsøy et al. [28] present in their study that the added roles lead to less autonomy for the team. However, this cooperation with surrounding roles is important for the team. In a study by Hoda and Murugesan [29], they present how a lack of clarity leads to a team's incomplete understanding of requirements, which leads to cascading problems, such as inaccurate estimation and possible rework.

An aspect of role interference related to requirements is goal setting and creating a shared vision. The ability to create a shared vision has been shown in research to be a key to success in software development since the beginning of the 1990s [30]. The importance of a clear, shared vision is also confirmed in recent studies on ASD teams at Spotify [31], and in a study on coordination in large-scale distributed ASD [32]. However, having a shared vision does not always happen as Moe et al. [22] describe in a study on a large-scale ASD transformation, where higher-level managers often defined goals that were not always seen as relevant or successfully communicated to the team. Defined goals were often treated the same as deadlines or deliverables and were not coordinated with other teams or with the rest of the organization.

Another threat to team autonomy is when external stakeholders steal resources [33]. For example, this could occur when customers or managers from various organizational departments approach developers directly with unrelated tasks and disrupt the work that teams are originally assigned to do. In such a situation, team autonomy is reduced [34, 15].

Finally, the large-scale frameworks may in themselves limit autonomy, as the need for structure at the inter-team level may come at the expense of individual team autonomy. For example, in organizations using SAFe, teams may feel limited by not being allowed to choose the sprint length, as well as by the use of pre-planning meetings [21]. However, little knowledge exists regarding the potential negative impacts of limited team autonomy in large-scale agile [10], and more studies are therefore needed.

2.4. Levels of team autonomy

In this study, we relied on the Authority Matrix that Hackman [35] developed as a theoretical approach to guide our analyses of changes to team autonomy in large-scale agile. This model describes four levels of team autonomy ranging from manager-led to self-managing, self-designing, and self-governing teams. As can be seen in Table 1, the amount of autonomy that teams are allowed is based on their responsibility and authority in four areas: (1) how to execute tasks, (2) how to monitor and manage work processes and progress, (3) how to design the team and its context, and (4) deciding on the overall direction [35]. A manager-led team is responsible only for decisions concerning how to execute their tasks. A self-managing (or self-organizing) team also monitors and manages work processes. A self-designing team has the authority to design and modify the team, as well as to make decisions regarding other resources (context). A self-governing team also decides on the team’s overall direction. The categories are not distinct types but rather show a continuum reflecting the increasing amounts of authority that teams hold relative to managers [35, 36].

Table 1. Levels of team autonomy [20].

Setting the overall direction				
Designing the team and organizational context				
Monitoring and managing work process and progress				
Executing team task				
	Manager-led unit	Self-managing unit	Self-designed unit	Self-governed unit

According to the Agile Manifesto, agile ASD teams should be allowed to decide *how* to build the product (i.e., executing tasks and managing work processes and progress), whereas the client decides *what* to build, thus contributing to setting the team's overall direction. and management should decide on the team's design and organizational context [17, 29]. In other words, self-organizing ASD teams commit to selecting and accomplishing their own tasks, as well as organizing themselves. Autonomous ASD teams should therefore be responsible for planning and scheduling their work, as allowing team members to participate in these activities will also increase their commitment to the teams' plans [33]. As such, autonomous ASD teams can be considered to be self-managing units. In this study, we are interested in understanding how autonomy changes when a large-scale agile framework is implemented. We will therefore return to autonomy levels in the discussion section.

3. Method

This study was based on an exploratory multiple-case study [37] of three cases called Agency, BizBank, and Vehicle that implemented large-scale agile development frameworks. The three cases of Agency, BizBank, and Vehicle were chosen based on a maximum variation approach [37], which involves selecting organizations that show variety in terms of size, business area, and corporate culture. The cases also had some similarities, which was important for the study and made them interesting to investigate. According to Yin [37], a case study research design is useful when "how" and "why" questions are investigated to achieve a rich and in-depth understanding of a phenomenon. Therefore, a case study approach was deemed appropriate in this study, as the purpose was to understand how autonomy is changing in large-scale agile software development. A multiple-case design was also preferred over a single-case design because the conclusions may offer more possibilities for suggesting generalizations [37].

The overall study investigated how inter-team coordination was performed and tailored over time in three organizations with disparate business logics. The unit of analysis was the 32 to 41 agile teams working in these cases. In all three of the organizations, the studied cases were the first parts of the organization in which SAFe was implemented. Agile ways of working were implemented, but the implementation of SAFe was novel to the organizations. Therefore, the change at each organization was not from a traditional project management approach to SAFe, as the organizations were already mature in agile ways of working. Also, this meant that no previous SAFe experiences existed in the surrounding organizations. Another important factor in the choosing of these organizations was timing. They all began to implement SAFe during the first part of 2017 and could therefore be studied in parallel. This also meant that they had similar external conditions, such as their worldwide economic situation, as they were all studied during the same time span. Therefore, data collected from these three cases were used, and data on the impacts of scaling up agile software development were further analyzed, focusing on the perceived impacts of SAFe on team autonomy.

3.1. Case descriptions

Agency is a pilot project formed by merging teams from two departments at a large government agency into one organization. The pilot project was aimed at discovering best practices for implementing SAFe roles, as well as the practices to be used within the entire organization. The software developed at Agency was web-based services for Swedish citizens. This involved developing backend functionality for administrators, as well as the frontend development of the web interface. The technical environment consisted of both mainframe and modern technical platforms. Teams were responsible for both the development and the operations of the services. Agency consisted of six teams from the start and ended up as nine cooperating teams. Team sizes spanned from four to 12 team members. Although the adoption of agile methods has been slow in the public sector [38], Agency was an early adopter that decided to implement agile ways of working in 2012. After five years of ASD, Agency decided to scale up by implementing the SAFe framework in a pilot project. A core team consisting of department managers and five agile coaches was formed with the purpose of organizing the implementation. The core team decided on all of the roles and oversaw the organizing of teams, which involved re-organizing some teams and team members.

BizBank is a software service development department in one of the largest business banks in Scandinavia. The department developed an in-house software service to manage client fund portfolios for administrators at the bank. This

meant backend functionality and system integration with other financial data services. Integrations with mainframe systems as well as modern technical platforms meant the need for different competencies within the teams. The teams were in charge of both the development and the operations of the developed software. The department grew from six to eight Scrum teams and decided to implement roles and practices based on the SAFe framework. The size of the teams spanned from five to 11 team members. For BizBank, the decision to begin working in an agile way was decided in 2014 when Scrum was implemented in various parts of the organization. In 2017, SAFe was implemented in one department of the bank consisting of five teams.

Vehicle is a department responsible for both the software and hardware development of a significant part of a motor vehicle. The department developed an in-house software platform and services for a motor vehicle. Because the department was also responsible for hardware development, the software platform continuously had to be in sync with changing hardware conditions. Teams were responsible for both the development and the operations of the software, but only for the development of the hardware. The department was first organized into 20 cross-functional teams, and in 2016, it began to implement roles and practices based on the SAFe framework. The 20 teams were organized into three separate units, and team sizes spanned from two to 13 team members. During the SAFe implementation, the department grew organically and ended up having 24 teams during the last visit of data collection. At Vehicle, the organization introduced agile ways of working by forming self-organizing teams in 2013 and decided to implement roles, processes, and practices that SAFe inspired in late 2016. Being a large department of more than 20 teams, it decided to scale up its work processes by forming three units of five to 12 teams each.

3.2. Data collection and analysis

A case study protocol was developed in line with the guidelines for the case study methodology that Yin [37] suggested to prepare for data collection. Semi-structured interviews were the most important source of information and were supplemented with the data collected from participant observations and a survey. Thus, multiple data sources were used for triangulation purposes as shown in Table 1. The general interview guide approach [39], also known as the semi-structured interview, was chosen for two reasons. One reason was to be able to adapt the questions based on the interviewees' roles and individual experiences. The other reason was that it made sure that the relevant topics were explored. The first draft of the interview guide was based on Heikkilä [40], who investigated teamwork in autonomous teams performing release planning. From this first version, the interview guide was constantly updated based on observations and insights from previous interviews. Appendix A shows a subset of questions from the last version of the interview guide (i.e., the questions most relevant to this study). Data collected through observations consisted of photos and field notes during the 17 visits to the case organization's premises. The field notes contained several discussions between employees in their daily work. Sentences from the discussions that were related to team autonomy were used as data.

Table 1. Data collection from case Agency, BizBank, and Vehicle.

Data source	Case Agency	Case BizBank	Case Vehicle
Number of on-site visits	5	6	6
Hours of observation	113	70	196
Interviews	6	8	14
Interview respondent roles	1 Agile Coach, 1 RTE 2 Scrum Masters, 2 Developers	1 Agile Coach, 1 RTE, 1 Product Owner, 1 Scrum Master, 3 Developers, 1 Software Tester	1 Manager, 1 RTE, 1 Product Owner, 3 Scrum Masters, 7 Developers

The on-site visits happened every second or third month and lasted for two to five working days. During these visits, interviews were performed, memoranda from meetings were taken, and a survey was distributed and collected. The survey questionnaire contained four sections of questions regarding teamwork and agile ways of working. Two of the questions used for this study were the following open-ended questions: "What do you consider to be the main benefit of

implementing SAFe in your organization?” and “What do you consider to be the main drawback of working according to SAFe in your organization?” Answers related to team autonomy were used as data in this study.

In total, 28 interviews were performed with key people (such as agile coaches and RTE’s) and team members who gave insights into multiple perspectives of the situation. The duration of the interviews spanned from 45 minutes to one-and-a-half hours. All interviews were recorded based on the participants’ consent, and the first author transcribed them. To ensure the interviewees’ anonymity, the respondents are referred to with role and case names throughout the paper. Data were collected from the beginning of 2017 to the end of 2018.

A thematic analysis method for data analysis was used by following the six steps that Braun and Clarke [41] presented: 1) familiarizing yourself with your data, 2) generating initial codes, 3) searching for themes, 4) reviewing themes, 5) defining and naming themes, and 6) producing the report. Thematic analysis is a method particularly suitable for systematically deriving themes across large amounts of data from different sources [41]. Themes are patterns that are meaningful in relation to the research question. In our case, the themes were related to changes in autonomy in the three organizations. The thematic analyses were conducted following a set of steps [41]. First, the interviews were transcribed and read several times so that we could become familiar with the data. The collected data were imported in NVivo, a tool for qualitative analysis. The next step, to generate initial codes, was performed by identifying patterns of perceived changes to team autonomy in the informants’ answers, observation notes, and answers to open-ended questions in the surveys. To guide our analysis, we used the theoretical lens of change in levels of autonomy [35].

Next, sections were labeled according to these patterns and were clustered into proposed themes. The first author and two other researchers, who also suggested initial themes, performed coding. The proposed themes were then reviewed, compared, and re-organized. The initial themes were then discussed in seminars with other researchers. Finally, we performed a final round of discussions and refining and ended up with six themes: *getting a better overview*, *making better long-term decisions*, *giving and receiving help*, *signaling limitations*, *limited feature choice*, and *enforced refinement*. The themes are presented in the result section.

3.3. Limitations

Inherent to any empirical study, this study has limitations. One such limitation relates to the reliance on interviews as the major data source. For the purpose of improving validity, respondents were chosen based on having enough experience with and knowledge of agile ways of working. Along with a large number of interviews (28), triangulation by observations, meeting memoranda, and a survey helped with reducing the study’s bias. For reliability to be ensured, interviews and observation notes were sent to employees so that they could check for errors or omissions.

4. Results

This section presents the six identified themes that describe perceived changes to team autonomy when SAFe was implemented. The themes are: 1) *getting a better overview*, 2) *making better long-term decisions*, 3) *giving and receiving help*, 4) *signaling limitations*, 5) *limited feature choice*, and 6) *enforced refinement*. We describe different areas, benefits, and drawbacks of these changes to team autonomy in detail in the various subsections. The first section, 4.1, presents the perceived overall view of changes to team autonomy.

4.1. Overall view of changes to team autonomy

Most of the interviewees stated that implementing SAFe had increased team autonomy. At Agency, a respondent stated that the organization was moving in the right direction: “*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*” (Agile Coach, Agency). A developer stated, “*Team autonomy has been affected a lot.... We didn’t really have autonomous teams, before but we have a real autonomous way of working now*” (Developer, BizBank). Most respondents at Vehicle also agreed. A Scrum Master expressed, “*I think that we are not as limited as we previously thought.... We’ve got more autonomy, I would say*” (Scrum Master, Vehicle). However, an engineer stated it had not changed: “*I would say that team autonomy is the same today as it was two years ago, before the implementation*” (Release Train Engineer, Vehicle). Several

respondents at BizBank also confirmed this. A Scrum Master said, *“No, I don’t think that [we have less autonomy]. I mean, the teams still decide how they do things.... We break down work and decide [on our own]”* (Scrum Master, BizBank).

At Agency, when the transformation had been going on for several months, the teams’ resistance to the SAFe implementation decreased because the team members were more actively involved. Surprisingly, the removal of agile coaches at Agency one year into the transformation resulted in increased team autonomy. When the agile coaches were not present, employees at Agency experienced more freedom to act independently and to make decisions regarding how to change their way of working without asking the agile coaches for approval. The teams reported perceiving more autonomy after the coaches had left. Several commented that they thought the agile coaches had enforced routines and processes from SAFe by the book without trying to adapt to the environment at Agency. A respondent stated, *“We can work more freely now when we don’t have any ‘method people’ around”* (Release Train Engineer, Agency).

Some respondents in all three cases expressed that team autonomy had not been changed following the SAFe implementation. At Agency, a developer expressed, *“What can I say about team autonomy? We don’t get to decide much now, as I see it, but we didn’t get to do that before, either.... I don’t think it has changed”* (Developer, Agency). A developer at BizBank confirmed the same view by stating the following: *“Nothing special. Think we get to decide most things just like before”* (Developer, BizBank). Finally, a Scrum Master at Vehicle expressed, *“The team has about the same mandate for decision-making now as they had before; no major difference there”* (Scrum Master, Vehicle).

To conclude, most of the respondents cited increased team autonomy in general, whereas a few expressed the status quo following the SAFe implementation.

4.2. Getting a better overview

Several respondents expressed that the benefits of implementing a large-scale agile framework included a better overview. This also meant more transparency in the project as well as an understanding of the bigger picture. This was addressed both in interviews and in 22 answers to the open-ended survey questions in all three cases.

At BizBank, one developer explained how the overall understanding increased due to joint planning and joint reviews. She stated, *“The knowledge of the whole business has definitely increased. You understand why you do things ... that you are part of a larger whole”* (Developer, BizBank). The developers perceived an increase in team autonomy because decisions on the team could now be based on an improved understanding of the overall picture as well as an understanding of what other teams were doing.

At Agency, respondents pointed out that a better overview improved teams’ ability to make better decisions. One informant expressed, *“The more we understand the whole and what the users of the system really need, we make better big picture decisions ourselves”* (Scrum Master, Agency).

As presented above, several respondents at BizBank did not perceive autonomy to be limited when scaling up. Rather, they perceived that team autonomy remained the same or was even improved. However, some team members seemed to have experienced less personal autonomy. According to one respondent, developers felt forced to participate in meetings, thereby limiting their personal autonomy:

“What has been quite tough for many is the change to be less isolated and having to listen to things in meetings that may not be interesting.... Many [developers] have a hard time with always having to share what they do. Getting everyone to understand the value of being transparent and listening to others all the time is something we struggle with” (Scrum Master, BizBank).

In other words, although some perceived participating in meetings as limiting personal autonomy, the impact on team autonomy offers a better overview, as sharing information makes the given work transparent to other teams.

4.3. Making better long-term decisions

Another area in which autonomy was perceived to have been changed was making decisions on a long-term basis. When one is working according to SAFe, an important practice is a joint planning session called PI planning, where plans are made for a number of future sprints. According to many respondents, planning several months ahead led to better long-term decisions. In addition to being addressed in interviews, this was addressed in 14 answers to the open-ended survey questions in all three cases. Understanding long-term plans for software products acted as feedback for the decisions that the team made. An informant stated: “[Planning] has had a feedback effect on us as a team. We have become very good at being foresighted and make better long-term decisions now” (Scrum Master, Vehicle).

A respondent at Agency stated: “It is easier to make our own decisions in the team now that we know where we are heading”(Scrum Master, Agency). In addition, decisions were more thought through due to this, according to a developer: “More thorough and overall better decisions now because we have a longer perspective than we had before” (Developer, Agency).

In another team at Vehicle, the Scrum Master expressed how he was fortunate to have more explicit long-term goals for his team. This made it easier for the team to be more autonomous when it came to making long-term decisions. He stated, “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” (Scrum Master, Vehicle).

However, teams at Agency also experienced resistance in making long-term decisions based on the organizational culture. An informant stated: “Whenever someone says that this has been decided, someone will right away ask: ‘Who has approved that?’ It is a culture of finding [one specific] person responsible for each decision” (Agile coach, Agency). The agile coach explained that this was a problem not only due to the organizational culture but also due to the rules and regulations within the organization.

4.4. Giving and receiving help

An area of changed autonomy that was perceived in all three cases was the ability to give and receive help more easily than before. This was brought up in interviews and discussions, as well as in 19 answers to the open-ended survey questions. A respondent at Vehicle said: “We have more autonomy when it comes to receiving help or helping others as well” (Developer, Vehicle). According to a respondent at Agency, this change was due to the previously expressed planning benefits of an improved overview and improved transparency: “They find it easier to help each other now. Before, they did not know what each team did, but now, they know it and can say, ‘We did a similar thing last sprint; come to us, and we can show you’” (Release Train Engineer, Agency). A developer at Vehicle confirmed this view, also pointing to an improved overview as a reason for the change, “It is easier to help out across team boundaries. Before, you didn’t really know what the other teams were doing, so you did not know that you could be helpful” (Developer, Vehicle). In a discussion during a PI planning meeting, a respondent even expressed that this change was a main difference related to team autonomy after the implementation of SAFe: “I don’t know if it [team autonomy] has changed really. Although, it’s evident that it is easier to help out between teams now, but nothing besides that” (Release Train Engineer, Agency).

Without pointing out any specific reason for this change, a respondent highlighted the increase in initiatives for helping other teams: “[It is] much easier to help out between teams. I see how often they take the initiative to ask for help and get help from each other during these two [planning] days” (Agile Coach, Agency). In particular, a recently hired employee pointed at this change as being especially important for the newly hired: “All teams are so helpful; it has been great for us who are fairly new. You only have to say that you do not understand something, and people from other teams have helped us” (Scrum Master, Agency). Helping other teams can often be done by presenting previous solutions. As a developer at BizBank expressed: “I notice that many share and help out more than they did before. Shows their own solutions and such” (Developer, BizBank).

4.5. Signaling limitations

Balancing one's capacity and workload is a challenge, and a change that the respondents expressed following the implementation of SAFe was the ability to signal limitations. In addition to being expressed in interviews and discussions, this was expressed in three answers to the open-ended survey questions at Vehicle and BizBank. A respondent at BizBank also highlighted an improvement in the ability to signal limitations to support team decisions regarding not adding work: *"[It has] really been an improvement to get this whole overview and get managers to actually see what we can do and cannot [do]"* (Developer, BizBank).

At Vehicle, a respondent expressed how the joint planning sessions made it easier to show and discuss actual capacity: *"One effect that has also come out 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"* (Manager, Vehicle). A developer at Vehicle who expressed a benefit of implementing SAFe also confirmed this: *"Easy to visualize workload versus resource"* (Developer, Vehicle).

Even when management wanted more from the teams, the transparent plans signaled capacity limitations and made it easy for teams to reject unrealistic demands. As one Scrum Master put it: *"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 ... that we could signal what limitations we had. We've got more autonomy, I would say"* (Scrum Master, Vehicle).

4.6. Limited feature choice

In all three cases, several respondents felt that implementing SAFe limited their ability to choose which features to implement. In addition to being addressed in respondent interviews, this was addressed in six answers to the open-ended survey questions at Agency and Vehicle. For example, in meetings among Product Owners at Agency, features were divided between teams without involving the Scrum Masters or the developers. The lack of involvement meant that teams did not understand enough about the features before the joint planning sessions took place. As a consequence, both the developers and the Scrum Masters were frustrated, and many of the features decided for the teams at Agency sometimes came as a surprise to the team members. A Scrum Master expressed: *"Features appear in the product backlog without the teams knowing about it. It is a pervasive problem. And it should not be that features pop up without us having a clue what it is"* (Scrum Master, Agency).

The limitation of choosing features was also put forth in two answers to the open-ended questions. The problem of not having the freedom to choose features was also evident at Vehicle. An informant complained that *"[Teams] 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 are already so detailed when they get them"* (Release Train Engineer, Vehicle). Another informant confirmed this view at Vehicle: *"Before, it was easier to choose whatever feature you wanted to work with.... We don't get to choose much now. It's much more dedicated [in advance]"* (Developer, Vehicle).

4.7. Enforced refinement

Because SAFe suggests a PI planning horizon of eight to 12 weeks, this means that teams need to understand enough about the upcoming features to be able to plan for such a period. In addition, it means that features need to be refined enough for the teams to understand and estimate them. Several respondents in all three cases explained that this need for understanding forced the teams into a greater amount of refinement than that which was needed before. This was also expressed in six answers to the open-ended survey questions. For example, a respondent at BizBank stated, *"We can decide very much on our own, but then, there is the issue of breaking down [features] a lot before planning"* (Tester, BizBank). A developer stated: *"We are forced into more refinement now than we were before"* (Developer, BizBank). Another respondent expressed that even though more refinement was enforced, it was not a big issue: *"Autonomy? It hasn't changed much. Sure, we are forced to do more refinement than we did before, but I don't see it as a problem"* (Developer, BizBank). A few people perceived the refinements as positive, which a developer at BizBank illustrated:

“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]” (Developer, BizBank).

However, the number of meetings required to agree on the refinements was seen as a major drawback. A developer stated, “We have so many preparatory meetings. They never seem to be satisfied with how much we refine before PI planning” (Developer, BizBank). The view of too many meetings was also expressed at Agency: “We’ve spent way too much time sitting in refinement meetings” (Scrum Master, Agency). At Vehicle, the impact of too much refinement was seen as stressful: “Very much time is spent on refinements, estimates and planning all the time. I think many people are like me and think that it can be quite stressful at times” (Developer, Vehicle).

5. Discussion

In this section, we discuss the research question *How does team autonomy change when a large-scale agile framework is implemented?* We used the Authority Matrix that Hackman [35] developed to guide our analyses of changes to team autonomy at different levels. The findings improve our understanding of the detailed impacts of implementing SAFe on team autonomy in large-scale ASD. This study also answers calls for future research on autonomous teams in ASD [5, 22]. Table 2 presents an overview of the changes in the levels of team autonomy for all themes.

Table 2. Changes in autonomy (increases or decreases) compared with Hackman’s levels of autonomy [35].

Themes	Areas of decision making			
	Task execution	Process monitoring and management	Designing team and context	Setting overall direction
<i>Getting a better overview</i>		Increased		Increased
<i>Making better long-term decisions</i>		Increased		Increased
<i>Giving and receiving help</i>	Increased		Increased	
<i>Signaling limitations</i>		Increased	Increased	
<i>Limited feature choice</i>			Decreased	Decreased
<i>Enforced refinement</i>	Decreased	Decreased		

A change to team autonomy was that respondents perceived to be *getting a better overview*. This change was identified in all three cases. Having a shared understanding of the organization’s overall goals and visions is an important key to success [30]. Team members expressed how they had obtained an improved overview of their work and of how their parts were important in the larger context, which enabled them to coordinate their decisions with the overall goals and visions. These findings confirm the importance of shared goals and activities in large-scale ASD [37, 32]. Looking at how a team’s level of autonomy changes based on Hackman’s [35] four areas, this theme suggests an increased influence of large-scale ASD on how to monitor and manage work processes and decisions with regard to the overall direction.

Another change to team autonomy that was identified in all three cases was the improved capability of *making better long-term decisions*. Per the investigation of how the level of team autonomy changed based on Hackman’s [35] four areas, this theme suggests an increased influence of large-scale ASD on how to monitor and manage work processes and decisions with regard to the overall direction. The PI planning practice, where several teams perform joint planning for a number of sprints [19], was specifically highlighted as an important practice. With the possibility of longer planning horizons, one could argue that there is a better chance for the team to decide on its own on how to achieve

these goals. This is consonant with Moe et al. [22], who discovered an increased commitment to their plans by letting teams decide more on their long-term plans and schedules.

A Scrum Master at Vehicle expressed that he or she was fortunate to have clear long-term goals, which was not the case for many other teams at Vehicle. The importance of having long-term goals for the team is highlighted in Šmite et al. [31], for example, but this often seems to be lacking on many teams according to previous research [22, 29].

In contrast to making better long-term decisions, an agile coach at Agency explained that some long-term decisions might be hindered due to an organization's in-house culture, where top managers need to approve decisions. This finding confirms previous studies on the importance of the company culture when agile is being scaled [20].

Another theme identified in all three cases was the improved capability of *giving and receiving help*. Team members expressed that the improved overview and the improved transparency due to joint planning with several teams made it easier for them to help other teams. The allocated time for joint planning with several teams made it easier to receive help from other teams as well. Regarding how the team's level of autonomy changed based on Hackman's [35] four areas, this theme suggests an improved influence on how to execute the tasks and context by helping other teams.

With a better overview from joint planning, 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 them out in advance. Moe et al. [22] presented a situation in which team members stated that they shielded their team from externalities, filtering out what they considered to be unnecessary for their team to take part in. In these three investigated cases, the teams instead reaped the benefits of not shielding their teams by being able to both help and receive help from one another.

Another theme identified in two of the cases, those of Vehicle and BizBank, was an improved capability of *signaling limitations*. When several teams conduct joint planning together, a high level of transparency of these plans might reveal which teams have too much work planned. The transparent plans therefore allow teams to signal capacity limitations more easily. Being able to signal limitations transparently is important because it might mitigate problems with interfering roles, such as stakeholders stealing resources [33], which is a problem in large-scale ASD [25]. Per the investigation of the level of team autonomy based on Hackman's [35] four areas, this theme suggests an improved influence of large-scale ASD on how to monitor and manage work processes and contexts, as work can be transferred to other teams.

An identified theme expressed in two cases, those of Agency and Vehicle, was *limited feature choice*. Team members expressed a limitation in choosing what to work with within the various teams. Per the investigation of how the level of team autonomy changed based on Hackman's [35] four areas, this suggests a decrease in autonomy, as a team in this situation becomes limited in terms of deciding the team's overall direction and context by not being able to have a say in what other teams should work on.

Dividing work between teams is part of deciding how to best accomplish the work in large-scale ASD [42]. At Vehicle, a respondent explained that some features just turned up in the product backlog and were not understood. This confirms the problems presented in Moe et al. [22] regarding higher-level managers defining goals not successfully communicated to the team. Involving teams in planning and goal setting is important because it increases motivation and a sense of meaningfulness [43].

The final theme, identified in all three cases, was *enforced refinement*. Refinement means that a team is involved in detailing the requirements for a specific feature together with product owners, product managers, or other stakeholders. Some employees felt that always attending refinement meetings was a struggle. They felt forced to comply despite not wanting to, which limited their autonomy. Regarding how the level of autonomy changed based on Hackman's [35] four areas, this suggests that a team becomes limited in terms of how to execute its tasks and how to monitor and manage work processes. This theme has a clear connection with the theme mentioned above concerning the limited possibility of choosing features. By conducting more refinement before each planning session with a specific set of features, a team was considered to be responsible for working with the features due to having gained more insights into

them. At BizBank, however, a respondent did not view the increased refinement as something negative. Rather, it was seen as an improved routine for preplanning, and it did not limit the possibility of choosing features.

5.1 Theoretical contributions

The findings from this study give us more detailed insights into the various levels of team autonomy that are affected when a large-scale agile framework is implemented. Previous research on the impacts on team autonomy in ASD, such as added roles leading to less autonomy [28], has not offered details into the areas of team autonomy that are being affected. An important theoretical contribution of this study is the detailed impacts on changes to team autonomy. This study suggests that all levels of autonomy are affected, and team autonomy is both increased and decreased in different areas of decision-making for agile teams.

The themes of *limited feature choice* and *enforced refinement* show a perceived decrease in team autonomy. However, these limitations in what to work with for the teams also create clarity regarding upcoming work for the teams. As Hoda and Murugesan [29] explain, a lack of clarity is an important problem with potentially cascading negative effects, such as possible rework. Previous research shows the importance of long-term clarity [31, 32], so perhaps these negative impacts on team autonomy are an acceptable price to pay to reap the benefits of a clear, shared plan. The notion that some level of autonomy must be sacrificed [4] makes sense in large-scale ASD because shared goals and visions are needed for effective coordination [30, 31, 32].

5.2 Practical implications

As described in the three themes of *making better long-term decisions*, *giving and receiving help*, and *signaling limitations*, the joint planning practice, PI planning, was viewed as important for these perceived positive changes to team autonomy. Dingsøyr et al. [28] found that added roles intended to clarify requirements and technical direction lead to less autonomy on teams, but they are necessary due to the possible impacts of a lack of clarity as found in Hoda and Murugesan [29]. This study suggests that a joint planning practice could mitigate the problem of a lack of clarity. Based on this, added roles might not be as important, or at the very least, they should not have as much of an influence on team decisions.

Along with a lack of clarity, another thing that these findings show is that a joint planning session might mitigate the problems of teams working in silos. Dickert et al. [1] present the problem of teams operating with differing priorities and agendas, and Moe et al. [22] explain how some teams shielded themselves [22]. The findings from this study instead suggest that teams are able to coordinate priorities as well as give and receive help due to joint planning. Strode et al. [24] present the importance of a synchronization strategy and suggest project synchronization once per project, as well as iteration synchronization once per sprint. These findings show the importance of implementing medium-range planning, planning a number of sprints ahead, per the theory of coordination [24] in large-scale ASD. Understanding the great importance and thereby possible benefits of the PI planning practice is an important practical contribution of this study when SAFe is implemented in an organization. For practitioners, we think this paper illustrates the importance of joint planning as well as a shared understanding of goals. The findings also put forth the importance of medium-range planning per the coordination theory in large-scale ASD.

6. Conclusion

This study presents several changes to team autonomy when a large-scale agile framework is implemented. Four identified themes showed changes that were perceived to increase the levels of team autonomy. First, teams experienced the benefits of *getting a better overview*, which suggests an increased influence on how to monitor and manage work processes and decisions with regard to the team's overall direction. Second, teams perceived an improved possibility of *making better long-term decisions*, which also suggests an increased influence on how to monitor and manage work processes and decisions with regard to the team's overall direction. The increased transparency and long-term planning might be reasons for this change. A better overview and increased transparency might also explain why many employees found that *giving and receiving help* was easier. This impact shows an improved influence of large-scale

ASD on how to execute the tasks and context by helping other teams. A practice that was specifically highlighted was the joint planning session, PI planning, which gave teams the capability of *signaling limitations* of capacity more clearly. When several teams planned and shared plans, a high level of transparency revealed which teams had too much work planned for themselves. This might improve the influence on how to monitor and manage work processes and contexts, as work can be transferred to other teams. Two themes were perceived to decrease different levels of team autonomy. One theme, supporting previous studies, was that teams experienced *limited feature choice*. With several teams needing to be coordinated toward a common goal, every team will not be able to choose freely. This suggests a decrease in autonomy because a team becomes limited with regard to deciding the team's overall direction and context. The last identified theme was *enforced refinements*, which suggests that the team becomes limited with regard to how to execute its tasks as well as monitor and manage work processes. The *enforced refinement* might also be a possible reason for the previously mentioned limited choice of features. With a certain team appointed for refining a set of features, the same team was often considered in advance to be responsible for the refined features.

For future work, an area mentioned in the discussion section is the differing views on *enforced refinement*, where some experienced them as a limitation when it came to team autonomy, whereas others expressed the associated benefit of better understanding upcoming features. These diverse experiences call for further research into the impacts of *enforced refinement* on team autonomy. Furthermore, some employees reported that the implementation of SAFe involved an increase in the amount of time spent in meetings. The high number of meetings was perceived as limiting personal autonomy, and the excessive meeting time was viewed as irrelevant for their work. When one is following large-scale frameworks, how to balance the need for coordination and transparency across teams with the need for teams to do focused work is worth investigating.

References

- [1] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87–108, Sep. 2016
- [2] A. Sablis, D. Šmite and N. Moe, "Team-external coordination in large-scale software development projects," *Journal of Software: Evolution and Process*, e2297, 2020.
- [3] J. H. Gundelsby, "Enabling autonomous teams in large-scale agile through architectural principles," *Proceedings of the 19th International Conference on Agile Software Development*, 2018, pp. 1-4.
- [4] J. M. Bass and A. Haxby, "Tailoring product ownership in large-scale agile projects: Managing scale, distance, and governance," *IEEE Software*, vol. 36, no. 2, pp. 58-63, 2019.
- [5] J.M. Bass, "Future trends in agile at scale: a summary of the 7th international workshop on large-scale agile development," In: *Hoda, R. (ed.) XP 2019. LNBP*, vol. 364, 2019, pp. 75–80. Springer, Cham.
- [6] N. B. Moe, T. Dingsøyr and K. Rolland, "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*, vol. 6, no. 3, pp. 45-59, June, 2018.
- [7] Digital.ai, *15th Annual State of Agile Report (2020)*, 2021. [Online]. Available: <https://digital.ai/resource-center/analyst-reports/state-of-agile-report>
- [8] M. Alqudah and R. Razali, "A review of scaling agile methods in large software development," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, no. 6, pp. 828-837, 2016.
- [9] I. Stojanov, O. Turetken and J. J. Trienekens, "A maturity model for scaling agile development," *Proceedings of the 41st Euromicro Conference on Software Engineering and Advanced Applications*, 2015, pp. 446-453.
- [10] Ö. Uludağ, P. Philipp, A. Putta, M. Paasivaara, C. Lassenius and F. Matthes, "Revealing the state-of-the-art in large-scale agile development: A systematic mapping Study," *arXiv preprint arXiv:2007.05578*, 2020.

- [11] R. Hoda, J. Noble, and S. Marshall, "Self-Organizing Roles on Agile Software Development Teams," *IEEE Transactions on Software Engineering*, vol. 39, no. 3, pp. 422–444, Mar. 2013.
- [12] R. A. Guzzo, and M. W. Dickson, "Teams in organizations: Recent research on performance and effectiveness," *Annual Review of Psychology*, vol. 47, pp. 307-338, 1996.
- [13] M. Fenton-O'Creivy, "Employee involvement and the middle manager: evidence from a survey of organizations," *Journal of Organizational Behavior*, vol. 19, no. 1, pp. 67-84, January, 1998.
- [14] B. L. Kirkman and B. Rosen, B, "Beyond self-management: Antecedents and consequences of team empowerment," *Academy of Management Journal*, vol. 42, no. 1, pp. 58-74. 1999.
- [15] C. W. Langfred, "The paradox of self-management: Individual and group autonomy in work groups", *Journal of Organizational Behavior*, vol. 21, no. 5, pp. 563-585, August, 2000.
- [16] A. Meier, M. Kropp, C. Anslow and R. Biddle, "Stress in agile software development: practices and outcomes," *Proceedings of the International Conference on Agile Software Development*, 2018, pp. 259-266.
- [17] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland and D. Thomas. (2015, June 11th). Manifesto for agile software development [Online]. Available: <http://www.agilemanifesto.org>
- [18] D. E. Strode, "A dependency taxonomy for agile software development projects," *Information Systems Frontiers*, vol. 18, no. 1, pp. 23–46, 2016.
- [19] D. Leffingwell, *SAFe 4.5 Reference Guide: Scaled Agile Frame-work for Lean Enterprises*. Addison-Wesley Professional, 2018.
- [20] M. Kalenda, P. Hyna, and B. Rossi, "Scaling agile in large organizations: Practices, challenges, and success factors," *Journal of Software: Evolution and Process*, vol. 30, no. 10, p. e1954, 2018.
- [21] H. Edison, X. Wang, and K. Conboy, "Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review," *IEEE Transactions on Software Engineering*, 2021.
- [22] N. B. Moe, B. H. Dahl, V. Stray, L. S. Karlsen, and S. Schjødt-Osmo, "Team autonomy in large-scale agile," *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS)*, 2019, pp. 6997-7006.
- [23] T. W. Malone and K. Crowston, "The interdisciplinary study of coordination," *ACM Computing Surveys (CSUR)*, vol. 26, no. 1, pp. 87–119, 1994.
- [24] D. E. Strode, S. L. Huff, B. Hope, and S. Link, "Coordination in co-located agile software development projects," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1222–1238, Jun. 2012.
- [25] M. Berntzen, V. Stray, and N. B. Moe, "Coordination Strategies: Managing Inter-team Coordination Challenges in Large-Scale Agile," in *Agile Processes in Software Engineering and Extreme Programming, Cham*, 2021, pp. 140–156.
- [26] A. Salameh and J.M. Bass, "Spotify Tailoring for Promoting Effectiveness in Cross-Functional Autonomous Squads," *Proceedings of the International Conference on Agile Software Development*, 2019, pp. 20-28.
- [27] D. Remta and A. Buchalcevova, "Product Owner's Journey to SAFe - Role Changes in Scaled Agile Framework," *Information*, vol. 12, no. 3, pp. 1-18, 2021.
- [28] T. Dingsøyr, N.B. Moe, T. E. Fægri and E. A. Seim, "Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation," *Empirical Software Engineering*, vol. 23, no. 1, pp. 490-520, 2018.

- [29] R. Hoda and L.K. Murugesan, "Multi-level agile project management challenges: A self-organizing team perspective," *Journal of Systems and Software*, vol. 117, no. 1, pp. 245- 257, 2016.
- [30] L. Gren, "Understanding work practices of autonomous agile teams: A socialpsychological review," *Proceedings of the International Conference on Agile Software Development*, 2020, pp. 227-235.
- [31] D. Šmite, N.B. Moe, M. Floryan, G. Lavinta and P. Chatzipetrou, "Spotify guilds: When the value increases engagement, engagement increases the value," *Communications of the ACM*, vol. 63, no. 3, pp. 56-61, 2020.
- [32] V. Stray, N. B. Moe, H. Vedal, M. Berntzen, "Using Objectives and Key Results (OKRs) and Slack: A Case Study of Coordination in Large-Scale Distributed Agile," In *Proceedings of the 55th Hawaii International Conference on System Sciences*, 2022.
- [33] N. B. Moe, T. Dingsøy, and T. Dybå, "Overcoming Barriers to Self-Management in Software Teams," *IEEE Software*, vol. 26, no. 6, pp. 20-26, 2009.
- [34] M. Hoegl and P. Parboteeah, "Autonomy and teamwork in innovative projects," *Human resource management*, vol. 45, no. 1, pp. 67, 2006.
- [35] J. R. Hackman, "The psychology of self-management in organizations". In M. S. Pallak & R. O. Perloff (Eds.), *Psychology and work: Productivity, change, and employment*, 1986, pp. 89-136.
- [36] C. C. Manz, "Self-leading work teams: Moving beyond self-management myths," *Human relations*, vol. 45, no. 11, pp. 1119-1140, 1992.
- [37] R. K. Yin, *Case study research and applications: Design and methods*, 6th ed. California, USA: SAGE publications inc, 2017.
- [38] J. Nuottila, K. Aaltonen and J. Kujala, "Challenges of adopting agile methods in a public organization," *International Journal of Information Systems and Project Management*, vol. 4, no. 3, pp. 65-85, April, 2016.
- [39] M. Q. Patton, *Qualitative research and evaluation methods (3rd ed.)*, SAGE Publication, 2002.
- [40] V. T. Heikkilä, *Case studies on release planning in agile development organizations*, Doctoral dissertation, Aalto University, 2015.
- [41] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77-101, 2006.
- [42] J. Sutherland and K. Schwaber. (2013). *The Scrum guide: The definitive guide to Scrum; The rules of the game* [Online]. Available: <http://www.scrum.org>
- [43] J. R. Hackman, "The design of Work Teams," in *Handbook of organizational behavior*, J. Lorsch, Ed. Englewood Cliffs, N. J., Prentice-Hall, 1987.

Appendix A. Interview protocol questions.

What is your role?

What does your team do?

How long have you been working here?

How long have you been working in an agile way?

Have you received any education/training for agile ways of working? If so, what was it?

How do you experience the introduction of large-scale agile processes in your organization?

- What has been difficult?
- What has been simple (easier than you thought it would be)?
- Has the introduction of large-scale processes had the impact you expected?
- Has it had any negative effects?

Describe how different forms of coordination are performed:

- PI Planning
- Scrum of Scrums
- Usage of the Program Board
- Communities of Practice

Has autonomy in your/each team changed? If so, in what way?

- In what areas do you have more control?
- In what areas do you have less control?

What do you think you could do (change) to increase autonomy for each group while still managing coordination effectively?

Biographical notes**Tomas Gustavsson**

Tomas Gustavsson is an Assistant Professor at Karlstad University. He has a Ph.D. in Information Systems from Karlstad University, an MBA from Stockholm University, and has many years of industry experience as an IT consultant. He is also an author of several prize-winning textbooks on agile development methods. His main research interests are agile ways of working, teamwork, coordination, and leadership in large-scale agile software development.

**Marthe Berntzen**

Marthe Berntzen is a Ph.D. candidate in Software Engineering at the Department of Informatics, University of Oslo. She received the M.Sc. degree from BI Norwegian Business School and has four years of industry experience. Marthe's Ph.D. research centers around inter-team coordination in large-scale agile software development. Her research interests include agile methods and practices, teamwork, leadership, and coordination in distributed settings. Her work has been presented at international conferences within software engineering, information systems and management and she has published research on leadership and coordination in distributed team settings.

**Viktoria Stray**

Viktoria Stray is an Associate Professor at the University of Oslo's Department of Informatics. She also holds a research position at SINTEF Digital. She has a Ph.D. in Software Engineering from the University of Oslo and an M.Sc degree in Computer Science from the Norwegian University of Science and Technology. Her research focuses on agile methods, autonomous teams, coordination, global software engineering, innovation, and large-scale development. She collaborates with large companies and provides research results on current industry-related topics. She has previously worked for Accenture in Norway's largest software development projects.