NTNU
Norwegian University of
Science and Technology

Inter-team coordination in large-scale agile software development

Anniken Østdahl

Master of Science in Computer Science
Submission date: June 2017
Supervisor: Torgeir Dingsøyr, IDI

Norwegian University of Science and Technology
Department of Computer Science

TDT4900 Master Thesis Computer Science

Inter-team coordination in large-scale agile software development

Author
Anniken Østdahl

Supervisor
Torgeir Dingsøyr

NTNU – Trondheim
Norwegian University of
Science and Technology

Spring 2017

Abstract

Agile approaches for software development have increased in popularity since the formulation of the Agile Manifesto in 2001. The methods were initially developed for small teams to be able to respond quickly to change and to be flexible in the development process. Many have also seen the benefits of close customer collaboration and having self-empowered teams. In the latest years, agile practices have also caught interest in larger organisations, to whether it can be applied to large-scale software development projects as well. However, the theoretical grounding for agile in large-scale is still limited.

As projects increase in size, there will be more teams and team members, which lead to increased complexity that urges the need for more coordination than in smaller agile projects. Inter-team coordination has thereby been identified as an important research topic within large-scale agile software development.
In this master thesis, inter-team coordination has been studied through an exploratory case study, on a very-large scale agile software development project, which was conducted in Norway. The data from this case were analysed with the Van de Ven model of coordination.

The primary results of this master thesis support many principles of the Van de Ven model of coordination. There are influential factors to coordination that demand different coordination mechanisms, and it was experienced in the analysed case, that these mechanisms and arenas for coordination changed over time. From more standardised, plan-based and formal coordination in the beginning, to be more informal and horizontal as the project progressed. The informal communication was seen essential to ease the inter-team coordination, by being able to make clarifications at lower levels. This was first possible when relations and trust, across the teams, were built. Another interesting finding was that large-scale projects need, to some extent, more standardisation and structure than smaller agile projects. However, the difficulty is to find the right balance, so the agility is not lost.

Keywords: **Agile, Large-scale development, Inter-team coordination, Van de Ven model of coordination, Scrum**

Abstract norsk

Smidige metodikker for programvareutvikling har økt i popularitet siden utformingen av manifestet for smidig utvikling i 2001. Prinsippene for smidig utvikling var opprinnelig ment for små team for å kunne reagere raskt på endringer og være fleksible i utviklingsprosessen. Mange har også sett fordelene med tett samarbeid med kunden, og ha selvstendige team. I de senere årene har større bedrifter fått interesse av smidige metodikker, og om det er mulig å bruke prinsippene også i større utviklingsprosjekter. Likevel er den teoretiske bakgrunnen for smidig utvikling i stor-skala fortsatt begrenset.

Når prosjekter blir større, blir det flere team og flere personer inkluderte som fører til større kompleksitet som krever mer koordinering enn med mindre team. Koordinasjon mellom de ulike teamene har blitt identifisert som et viktig forskningsområde innen smidig utvikling i stor-skala. I denne masteroppgaven har det blitt gjennomført et utforskende case-studie på et stor-skala smidig prosjekt i Norge hvor fokuset for analysen har vært på koordinering mellom team. Dataen fra prosjektet har blitt analysert med Van de Ven modellen for koordinasjon.

Hovedfunnene i denne masteroppgaven støtter opp under flere av prinsippene i koordineringsmodellen Van de Ven. Det finnes flere påvirkende faktorer som krever ulike mekanismer for koordinasjon, og det var erfart i det evaluerte prosjektet at mekanismene og arenaene for koordinering endret seg over tid. I begynnelsen var det opplevd at det var mer standardisert, formelt og bruk av planer, mens etter hvert ble det mer uformell koordinering på tvers av prosjektet. Den uformelle kommunikasjonen var essensiell for å gjøre koordineringen mellom team bedre slik at mange utfordringer kunne bli løst på team nivå. Dette var først mulig når relasjoner og tillit var opprettet mellom team medlemmene. Et annet interessant funn er at stor-skala smidige prosjekter trenger mer standardisering og struktur enn mindre smidige prosjekter. Derimot kan det være vanskelig å finne en rett balanse mellom ordnede former og fortsatt bevare smidigheten til prosjektet.

Preface

This master thesis is written Spring 2017 as part of the Master program in Computer Science at the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU).

My specialisation within Computer Science has been software development, and through several subjects at NTNU the processes and methodologies of software development has caught my interest. Thereby I choose to research agile methods at large-scale for my master thesis as it is quite a new field of research, and also an interesting and important area. I choose to look closer into inter-team coordination for this master thesis as that stood out in the preliminary project to be a major factor for agile in large-scale projects; however, there was a gap in the theoretical grounding which I wanted to study closer.

I would like to thank Torgeir Dingsøyr for being a helpful and motivating supervisor for both the preliminary project (Fall 2016) and this master thesis. His knowledge and experiences within the field of agile methods have been really insightful, and I have appreciated having meetings evenly, but still be able to do the work at my own pace.

Anniken Østdahl
Trondheim, June 13, 2017

Contents

I Introduction 13

1 Introduction 15
- 1.1 Motivation 15
- 1.2 Problem description 16
- 1.3 Scope and limitations 16
- 1.4 Target audience 17
- 1.5 Report outline 18

II Theory 19

2 Agile software development 21
- 2.1 Introduction to agile software development 21
- 2.2 Scrum 24
- 2.3 Large-scale agile development 25
  - 2.3.1 Taxonomy of scale 26
- 2.4 Scrum-of-Scrums 27
- 2.5 Scaling frameworks 28

3 Coordination 31
- 3.1 Introduction to coordination 31
  - 3.1.1 Coordination in large-scale projects 32
- 3.2 Van de Ven model of coordination 33
- 3.3 Impersonal mode of coordination 35
- 3.4 Personal mode of coordination 36
- 3.5 Group mode of coordination 38
- 3.6 Van de Ven hypotheses 39

III Research methodologies 41

4 Method 43
- 4.1 Literature review 43
- 4.2 Research method 44
  - 4.2.1 Case selection 45
  - 4.2.2 Data collection 45
  - 4.2.3 Data analysis 45
- 4.3 Quality of case study 47

IV Results and evaluation 49

5 Results 51
5.1 The Omega Project 51
5.2 Van de Ven model 55
5.3 Impersonal mode of coordination 58
5.3.1 Task uncertainty 59
5.3.2 Task interdependence 59
5.3.3 Unit size 59
5.4 Personal mode of coordination 60
5.4.1 Task uncertainty 62
5.4.2 Task interdependence 63
5.4.3 Unit size 63
5.5 Group mode of coordination 64
5.5.1 Task uncertainty 66
5.5.2 Task interdependence 66
5.5.3 Unit size 67

6 Discussion 69
6.1 Impersonal mode of coordination 70
6.2 Personal mode of coordination 71
6.3 Group mode of coordination 73
6.4 Evaluation Van de Ven model 74
6.5 Evaluation research questions 75
6.6 Limitations of research 78

7 Conclusion 81

Bibliography 83

List of Tables

1.1 Report outline 18

2.1 Traditional vs agile software development Nerur et al. (2005) 23
2.2 Taxonomy of scale of agile software development projects (Dingsøyr et al., 2014) . . 26

3.1 Impersonal mode of coordination (Dietrich et al., 2013) 35
3.2 Personal mode of coordination (Dietrich et al., 2013) 37
3.3 Group of coordination (Dietrich et al., 2013) 38

4.1 Databases for literature search 44
4.2 Participants in focus groups 45

5.1 Roles in scrum teams (Dingsøyr et al., 2017) 54
5.2 Coordination arenas and methods 56
5.3 Countiuned from Table 5.2 - Coordination arenas and methods
2:*Twitter is an online news and social networking service where users post and interact with messages, "tweets", restricted to 140 characters* 57

List of Figures

2.1 Scrum . . . 24
2.2 Scrum-of-Scrums scaling . . . 27

5.1 Division of teams in Omega project . . . 52
5.2 Location of teams . . . 53
5.3 Hierarchy of scrum meetings . . . 55

Part I

Introduction

Chapter 1

Introduction

This chapter will in Section 1.1 describe the motivation for conducting this study regarding large-scale agile software development, and why the topic inter-team coordination was chosen to study more thoroughly. Then, in Section 1.2 the problem description will be presented, as well as the research questions that have been the foundation for this study. In Section 1.3 scope and limitations of the study will be presented before the target audience in Section 1.4 and an outline of this report in Section 1.5.

1.1 Motivation

Agile development methods have increased in popularity, and in software development today there is a need for rapid changes and to have an optimised work culture with effective collaboration and communication.
Agile methods have proven to be successful in software development in small development teams, and have become a very attractive methodology (Cohen et al., 2004). Some of the reasons for its popularity are the ability to respond to change, self-organizing and competent teams, close collaboration with customer and continuous delivery of working software (Greer and Hamon, 2011).
The software development community has embraced agile methods because of face-to-face communication, technical excellence, simplicity, and the continuous adaption (Lee and Xia, 2010). The evolutionary delivery with iterative cycles with intense collaboration has been successful in practise (Nerur and Balijepally, 2007). In these cycles, the active involvement of customer/stakeholders has led to closer cooperation and more satisfied customers. It has also been stated that the customers felt they had more control over the process which is beneficial for both parties (Dingsøyr et al., 2012, Dybå and Dingsøyr, 2008).

The last years, there has been an increasing interest to adapt agile methods in larger and more complex projects, as one has seen the benefits of agile methodologies (Dingsøyr and Moe, 2013). Many agile principles have proven to be most satisfactory to small projects and small teams. Therefore there has been an increasing interest in studying how agile methods scale to larger projects. Agile software development needs a different approach to planning, control and coordination, and especially the inter-team coordination needs to be handled differently when one is dealing with more than one agile team (Dikert et al., 2016, Tore Dybå and Moe, Tore Dybå and Moe).
At the International Conference on Agile Software Development (XP2013), *inter-team coordination* was voted as a high-priority topic in large-scale agile software development (Dingsøyr and Moe, 2013). Inter-team coordination has been identified to be a crucial factor in large-scale agile development in

the preliminary research project (Østdahl, 2016) and thereby caught an interest to study further. It has also been identified through several studies to be an important factor to succeed with large-scale agile (Larman and Vodde, 2010, Scheerer et al., 2014). Inter-team coordination is a challenge that comes when scaling agile methods. With more individuals and teams, there is a need for more coordination since the overall complexity increases as size increases (Scheerer et al., 2014). However, the theoretical grounding, through empirical research studies, is still limited (Scheerer et al., 2014). That is one of the reasons for choosing this topic, to be able to gain more insight and to contribute to the agile at scale research community.

1.2 Problem description

In the lack of understanding for how autonomous teams, in large-scale agile projects, can most efficiently and successfully coordinate across the teams, this master thesis will try to contribute to increasing the insight of inter-team coordination. Thereby, this master thesis aim at answering:

> **How can inter-team coordination be achieved in a very-large scale agile software development project?**

The reason for this main research question is to look further into the topic of inter-team coordination by conducting an analysis of a case to gain a deeper understanding of the topic. The analysed case, were of very-large scale agile, meaning it was more than ten agile teams involved. It was thereby very well suited to analyse regarding inter-team coordination. The chosen case practised Scrum as a methodology during the project.
Two sub-goals were also identified, to be able to clearer vision what aspects that interfere with coordination by looking at hindrances.

> **Sub-goal 1:** What are the challenges, and the consequences of these?
> **Sub-goal 2:** How can one deal with these challenges?

By conducting research with the outlined goal in mind, and substantiate with the sub-goals, it should be possible to identify practices used to achieve inter-team coordination, as well as challenges that are noticed through a real case.

1.3 Scope and limitations

As the master thesis is constrained by time, the scope of the project was also needed to be adjusted to be able to produce results within the time restrictions. As the researcher did not perform the interviews, but received focus group material to be analysed, in collaboration with Agile 2.0, more resources and time were spent on conducting a thorough analysis. The results of this research are based on what could be found by framing the analysis phase by using a model of coordination. Since the analysis was conducted using a particular model, the results will also be influenced by that. The results will aim at comparing the model against what could be observed in the analysed case. Also, the discussion of the results will focus on the research questions, found in Section 1.2, to evaluate whether the chosen model is suitable to assess inter-team coordination.

This master thesis does not aim at introducing new theory regarding inter-team coordination and agile practices at scale. The purpose of this study is to gain a deeper knowledge and insight of the research area, and not revolutionise and present new ways of successfully practising inter-team coordination.

The background theory of agile software development and coordination is included in this master thesis to give a broader overview and good insight of the topics that are essential for the analysis. They will not be elaborated endlessly, but focus on giving a brief insight of what studies have been conducted and what the theoretical grounding for these topics are.

As the researcher conducted a preliminary project within the subject of large-scale agile software development, some of the background theory, on agile software development, is included or rewritten for this purpose (Østdahl, 2016). The reason for this that was much valuable work was done, and data gathered in the preliminary study. Thereby, some parts just needed to more thorough and elaborated a bit more.

1.4 Target audience

There are several who could have an interest in this research, to gain a deeper understanding of inter-team coordination. This master thesis is targeted at agile communities, and thereby researchers within the field of agile at scale. They would most likely find this research interesting and valuable, as there are not that many contributions yet, within the field.

Another target audience might be those who practise agile methods in large-scale or are thinking of applying agile methods to their organisation. This study will give valuable insight on both benefits and challenges with inter-team coordination.

The last group that might have an interest in this research is students, and especially those who are studying topics related to computer science and software development in general. It might be more interesting to read insight and findings from a real case, than outdated textbooks. The practices of software development are changing rapidly, and as new techniques and methods are constantly evolving, school textbooks are unfortunately fast outdated. It might also be more interesting to see a case with how methods are applied in practice and not only how textbooks describe it to be.

1.5 Report outline

In Table 1.1 the different chapters and what they include are presented to give an overview of what the rest of this paper will consist of.

Chapter 1 : Introduction	This chapter include motivation for conducting this research as well as the overall problem description for the master thesis.
Chapter 2 : Agile software development	Include background theory on agile software development, as well as theory on agile at scale. The common agile methodology Scrum will be presented, as well as the technique Scrum-of-Scrums. Lastly, a brief overview of multi team systems are presented.
Chapter 3 : Coordination	First some general coordination theory is presented, thereafter some different coordination theories. Next there will be some insight on coordination in large-scale projects and inter-team coordination from other studies. And last the Van de Ven model of coordination will be presented.
Chapter 4 : Method	Describes how the exploratory case study was conducted with reasoning of chosen model for framing the analysis.
Chapter 5 : Results	First there is a presentation of the Omega case. Then the findings from using the Van de Ven model on that case is presented.
Chapter 6 : Discussion	An evaluation and discussion of the insight gained from the analysed case will be presented. Thereafter, the results will be evaluated against the Van de Ven model. Lastly a discussion in relation to the research questions in Chapter 1 will be presented.
Chapter 7 : Conclusion	A conclusion based on the findings in the exploratory case study

Table 1.1: Report outline

Part II

Theory

Chapter 2

Agile software development

In this chapter some general theory on agile software development will first be presented to give insight within the field in Section 2.1, and what kind of research that has been conducted earlier. Then, in Section 2.2 one of the most common agile methodologies will be presented, namely Scrum. After that, some previous studies on large-scale agile development will be presented in Section 2.3. In Section 2.4, the scaling technique Scrum-of-Scrums will be presented, before some other scaling frameworks will be mentioned lastly in Section 2.5.

2.1 Introduction to agile software development

The software development processes today are affected by change, and of project requirements that often are unclear, and maybe even unknown. This state the need for more agile development methodologies (Rising and Janoff, 2000). Already, in the late 1990's, several methodologies came along with a different combination of old and new ideas. These new methods had in common that they emphasised tighter collaboration between the development team and the business stakeholders, and embraced changing environments.

The agile manifesto was formulated in 2001, which states the following principles for new ways to uncover valuable strategies for software development[1]:

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

Some of the characteristics with agile methods are the value of competent people within the team who has knowledge and relations that are valuable contributions to software development. Also, evolutionary delivery through short iterative cycles - of planning, action, reflection - intense collaboration, self-organizing teams, and a high degree of developer discretion are all aspects of agile practices (Nerur and Balijepally, 2007). Agile methods actively focus on involving the customer from the beginning of the process and facilitate feedback and reflection. This makes the development process more likely to end up with a satisfying result (Dingsøyr et al., 2012).

[1] https://www.agilealliance.org/agile101/the-agile-manifesto/ (2001)

Agile methods are a new way of thinking, compared to the traditional development methodologies, which mean it will impact the organisational structure, culture, and management practices. This will possibly lead to several challenges in the transition to agile methods. To successfully adapt to agile methodologies the organisation must rethink their goals and reconfigure human, managerial, and technology components (Nerur et al., 2005). While traditional methods focus on stricter management and rely on up-front planning, agile methods aim at accepting and efficiently manage change (Dikert et al., 2016).

Highsmith summarises agile approaches to be questioning assumptions, and encourage change. He states the importance of flexible planning, extensive collaboration, and learning to be able to achieve agility in a project (Highsmith, 2002). Since agile methods are emphasising less documentation, it gives the flexibility to respond to changing conditions (Serrador and Pinto, 2015). Lee & Xia (2010) define the team's ability to respond to changing requirements, efficiently and effectively, as software development agility. This agility is possible by having short, incremental, iterative development cycles, self-organizing teams, active involvement of stakeholders, and continuous delivery.

With agile teams, thorough collaboration and communication among team members are essential to provide a basis for collective action. The team members are empowered with the possibility to contribute to decision-making and are not restrained to a particular specialised role. That increases the diversity of the teams and makes it possible to self-organize and provide a team that can respond quickly to situations in a changing environment (Nerur et al., 2005). One of the characteristics of agile teams, which Dybå et al. refer to, is that *'they have faith in their abilities, show respect and responsibility, establish trust, and preserves the quality of working life'* (Dybå and Dingsøyr, 2008). This describes the foundation for good agile practices, where the team members are motivated and want to contribute to accomplishing good results. That is possible when the team members are self-organized and empowered.
Agile development methods recognise the team members competencies, and that the appropriate skills and empowering are crucial in efficient decision-making. By having autonomous teams, it facilitates creativity in problem-solving as well as efficiency (Lee and Xia, 2010). Diverse skills and perspectives trigger learning and innovation, which can lead to better and more efficient solutions to complex challenges. However, this is also dependent on how well the team members communicate and collaborate, as it can be more challenging with diverse teams who have different points of view, which again may cause conflicts.

Agile methodologies differ in many ways from the traditional development methodologies, and in the Table 2.1, some of the most important differences are listed. These differences are worth noticing as they are quite the opposite approaches to dealing with several parts of a project. Agile methodologies are more adaptable to changes and rely heavily on the teams being self-empowered. Traditional methodologies are more formal and planned from the beginning to the end of a project.

	Traditional	**Agile**
Fundamental Assumptions	Systems are fully specifiable, predictable, and can be built through meticulous and extensive planning	High-quality, adaptive software can be developed by small teams using the principles of continuous design improvements and testing based on rapid feedback and change
Control	Process centric	People centric
Management Style	Command-and-control	Leadership-and-collaboration
Knowledge Management	Explicit	Tacit
Role Assignment	Individuals - favours specialization	Self-organizing teams - encourages role interchangeability
Communication	Formal	Informal
Customer's Role	Important	Critical
Project Cycle	Guided by tasks or activities	Guided by product feature
Development Model	Life cycle model (Waterfall, Spiral etc)	The evolutionary-delivery model
Desired Organizational Structure	Mechanistic (bureaucratic with high formalization)	Organic (flexible and participative encouraging cooperative social action)
Technology	No restriction	Favours object-oriented technology

Table 2.1: Traditional vs agile software development Nerur et al. (2005)

Several agile development methodologies have been developed since the introduction of the agile manifesto - like Scrum, XP, Crystal methodologies, Feature-Driven Development (FDD), etc. They all share a more collaborative development, "lean" mentality[2], close collaboration with the customer, and an acceptance that uncertainty is a part of software development. Each method has its strengths and weaknesses and is thereby suitable for different kinds of projects (Nerur et al., 2005).

In the next section, one of the agile methodologies will be presented; namely, Scrum, which is one of the most commonly used agile methods (West et al., 2010).

[2] Lean mentality means in this context: minimising unnecessary work

2.2 Scrum

Scrum is an iterative and incremental software development process that is designed to respond to changes quickly. It is mainly suited for small teams, where it should not be more than ten members of a team to enable effective communication and fast feedback. The team works tightly together, and an important factor is to have a single focus; therefore, the self-organizing team must have clear priorities. It is a favoured agile methodology as is emphasised how people work and not the work they do (West et al., 2010).

A model of how Scrum is commonly practised can be found in Figure 2.1. First, you usually have an initial planning phase where the project team develops an architecture and identifies a chief architect. It is important to define a project vision based on this, and the chief architect should ensure the visions are consistent throughout the development phases (Rising and Janoff, 2000).
In Figure 2.1 [3], the first element is the product backlog. In the product backlog, all the identified tasks are kept in a prioritised list, which is usually done in the initial phase. There, a high-level description of all the requested features for the project can be found. For each sprint, tasks are chosen from the product backlog and put in the sprint backlog during the sprint planning meeting. The sprint backlog consists of the identified tasks from the product backlog, which have been decided to complete in the next sprint.

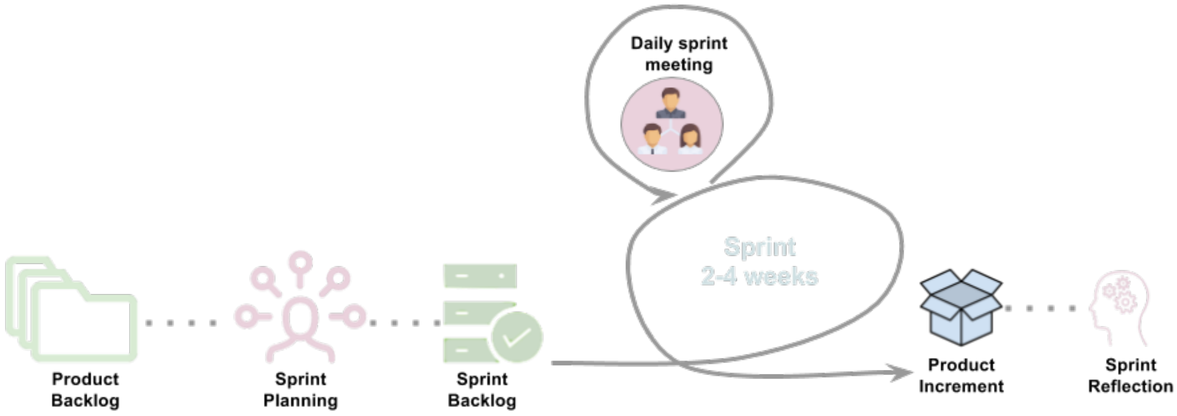

Figure 2.1: Scrum

After the initial planning, the next phase of the project it is divided into short development phases called sprints. The goal of a sprint is to deliver a product increment. A sprint usually lasts from one to four weeks. The key idea of a sprint is to provide valuable functionality at the end of a sprint, and each of these products increment builds on previous increments (Rising and Janoff, 2000).
During a sprint, it is common to have daily stand-up meetings where you involve all the team members and get a quick update what has been done, and what the plan of the day is. This session work as a team-building purpose as it makes your work visible to others as well as feeling part of a group. These meetings serve as keeping everyone informed of team progress and obstacles, and gives the opportunity to solve challenges as quickly as possible.
The team also has to decide a scrum master. A scrum master should work as a motivator and

[3] https://en.wikipedia.org/wiki/Scrum_(software_development) *Accessed:12.12.16*

coordinator for the rest of the group by ensuring that everyone makes progress, record the decisions made and track the actions being done. The scrum master should also ensure to keep the scrum meetings short and focused[4].

At the end of a sprint, the team shall have produced a valuable increment that builds on earlier increments of the project. It is common to have a retrospective at the end of each sprint for several purposes. First, the stakeholders should be involved so that work can be added, eliminated or re-prioritised. Second, the retrospective is also good for the team spirit, where the team can evaluate what went well, and what could be improved for the next sprint.

Scrum is often adjusted and tailored a bit when used in practice to make it natural for the teams to apply the methodology. This has much to with not feeling restrained by having to do all practices of the methodology just because one are "supposed to". Several use the methodology as a starting point and tailor it to suit the project at hand, as well as tailor it to the team members that are practising it. By forcing on several practices which the team does not feel suits their work, will most likely lead to frustration and inefficient use of time. However, there have also been reported several cases where agile has been customised poorly, where an organisation has tried to tailor to meet specific needs. This has resulted in several essential practices being dismissed which also has led to problems (Dikert et al., 2016).

2.3 Large-scale agile development

In the more recent years, there has been a change that contradicts to common agile myths about small, co-located teams that are able to scale (Rico, 2010). Larger projects and organisations have noticed that agile methods are more flexible and adaptable to change (Kude et al., 2014), and has become appealing methods to adapt in larger organisations which are seeking to improve their performance (Dikert et al., 2016). Within the field of software development, adapting agile methods have been widely recognised (Olsson and Bosch, 2015). Agile development is needed to keep up with the constant demand for change and innovation, also in large-scale. One need to realise that unpredictabilities play a more prominent role than before, which makes it suitable to use agile methods which are more adaptable to change (Laanti, 2014). However, as Saaeda et al. (2015) point out, applying agile at scale face several challenges e.g. too many meetings, lose sight on the big picture and lack of schedule. Scheerer et al. (2014) mention difficulties with large-scale projects when it comes to *'agile's desired organic structure, tacit knowledge management and informal communication'*. These aspects are found more difficult in large-scale.

Lopes (2014) states that large-scale agile software development requires a need for organisational cultural change, both at a team level, and high management level. Also, Dybå et al. (2008) point out that introducing agile methods to a larger organisation, increase in difficulty as organisation increase in size. This is mainly because a bigger project also involves more people that all need to be on the same page, regarding methodology, to make it work. If there are not a shared understanding of the project and work habits, one might meet several challenges during implementation that could have been avoided.
With more team members and teams, the project increases fast in complexity (Bick et al., 2016). Larger agile projects are thereby characterised by the need for coordination (Dikert et al., 2016). One of the main reasons it is more challenging with agile in large-scale is what differs from small scale agile; namely, there will be more dependencies, both between different sub-projects and between the teams

[4] https://www.mountaingoatsoftware.com/agile/scrum *Accessed: 25-09-2016*

(Lindvall et al., 2004). That urge the need for more standards and structure, which thus reduces the agility (Saeeda et al., 2015). Scaled agile also need different practices for handling reflection towards work habits on how to become more efficient, which is solved through retrospectives in small agile, but new mechanisms are needed to coordinate all the teams in large-scale agile (Bjørnson and Vestues, 2016). Eckstein (2016) identify that one of the core values of agile practices; namely, self-organising teams, will also be more difficult in large-scale agile, and have suggested a model to help to handle that.

Scaled agile have been a hot research topic in the agile community for some years now (Dingsøyr and Moe, 2013). And it has been a topic at the International Conference on Agile Software Development the latest years (*XP2013, XP2014 and XP2015*) (Bjørnson and Vestues, 2016, Dikert et al., 2016). However, there are not that many studies that have been conducted, and there is a lack of theoretical grounding (Dingsøyr et al., 2014, Serrador and Pinto, 2015). Some topics that have been pointed out that are important in large-scale agile after XP2016 are: distributed large-scale, inter-team coordination, knowledge sharing, large-scale agile transformations and multidisciplinary work (Moe et al., 2016).

Several fundamental assumptions of agile development are tested, and there are some limitations in applying agile methods on large-scale projects (Dingsøyr and Moe, 2014, Saeeda et al., 2015). For instance, inter-team coordination is non-agile in nature as the teams are supposed to have self-management and be empowered. With large-scale agile, there is a need to coordinate with other teams effectively which can be challenging. Some have also mentioned the difficulty in balancing between being a member of a specific team in relation to their membership to the project as a whole (Moore and Spens, 2008).

As large-scale agile development is quite a new research field, there are, as mentioned, not that much empirical research being conducted yet. However, the topic is of increasing interest, and many are curious about knowing how agile methods can be applied with success in large-scale projects.

2.3.1 Taxonomy of scale

As large-scale agile have increased in interest, Dingsøyr et al. (2014) summarise several articles with different definitions of what large-scale agile development can be defined to be, as there are several interpretations of the buzz-word (Dingsøyr et al., 2014). They state that there was a need for a standard taxonomy of scale. Throughout this research paper, that taxonomy of scale will be used, which is described in Table 2.2.

Level	**Number of Teams**	**Coordination approaches**
Small-scale	1	Coordinating the team can be done using agile practices such as daily meetings, common planning, review and retrospective meetings
Large-scale	2-9	Coordination of teams can be achieved in a new forum such as a Scrum of Scrums forum
Very-large-scale	10+	Several forums are needed for coordination, such as multiple Scrum of Scrums

Table 2.2: Taxonomy of scale of agile software development projects (Dingsøyr et al., 2014)

2.4 Scrum-of-Scrums

Scrum, which is presented in Section 2.2, is a commonly used agile methodology and is a representative methodology for agile methods. Scrum is initially best suited for a small team with fewer than ten members. However, applying Scrum in large-scale projects has been of increasing interest, and several mechanisms for handling inter-team coordination have been explored and researched. One of the few mechanisms for scaling the methodology for larger projects, and dealing with multiple teams, is Scrum-of-Scrums[5](Paasivaara et al., 2012, Schwaber, 2004). Scrum-of-Scrums is defined as

> *'An approach to coordinating the work of multiple Scrum teams wherein one or more members of each Scrum team come together to discuss and resolve inter-team dependency issues'* [6]

Scrum-of-Scrums meetings are one of the few known practices for handling inter-team coordination, and is illustrated in Figure 2.2 [7].

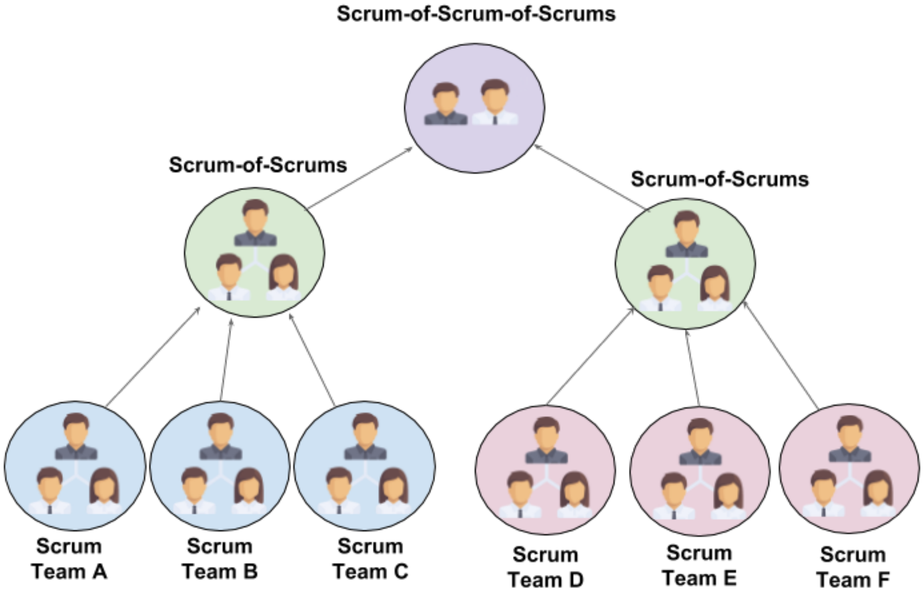

Figure 2.2: Scrum-of-Scrums scaling

Scrum-of-Scrums meetings are similar to daily the Scrum meeting; however, the main difference is that it involves teams, instead of team members. This brings challenges to keep it interesting for everyone, as well as effective. Also, the Scrum-of-Scrums meetings should be used as a way of keeping the teams synchronised and coordinated, and not end up in being status report meeting for the management (Larman and Vodde, 2010). Commonly, each team sends one representative to the meeting, and this person may change over time since different expertise from the team is needed at various times of the project time. The person, the team sends, should be the one who is in the best position to understand and respond to issues (Cohn, 2007).

[5] Scrum-of-Scrums is also called meta scrum by some researchers

[6] http://www.innolution.com/resources/glossary/scrum-of-scrums-sos *Accessed: 14.02.17*

[7] https://www.agilest.org/scaled-agile/scrum-of-scrums/ *Accessed: 14.02.17*

Scrum-of-Scrums can be divided up even further, as seen in Figure 2.2, where Scrum-of-Scrum-of-Scrums is also included. However, this technique, by scaling up too many levels, has been discussed whether it is beneficial or not.

There are different views on how long, and often the Scrum-of-Scrums meetings should be, and this can also be crucial whether it will be successful or not. Schwaber (2004) suggests having meetings every day, like daily meetings in Scrum, while Cohn (2007) suggests to have longer meetings, but not that often like two to three times a week.

Paasivaara et al. (2012) have studied two different projects where Scrum-of-Scrums have been applied. Their findings state that Scrum-of-Scrums meetings are affected by having too many participants that make the meetings inefficient since it is affected by different interests and concerns. Also, it is experienced that it can become very time-consuming. However, they saw the benefit of having smaller, focused inter-team meetings with participants of similar goals and interests (Paasivaara et al., 2012).

The efficiency and result of Scrum-of-Scrums are dependent on how the meetings are structured and organised, and should be used to ensure the coordination and integration of output from the teams [8]. Emphasis should be put on coordination as well as solving the teams' barriers. The intention of these meetings is to make sure each team achieve their sprint goals, and that the primary goal of the project is met by all teams.

The Scrum-of-Scrums technique fits naturally into Scrum and promotes many agile and lean values, as well as principles like self-organizing and empowering of the teams [9].It is also a good starting point for coordination to get a feeling of what the other teams are doing since the teams have to coordinate somehow anyway. With scheduled meetings, it helps to identify situations across the teams that need coordination. However, one of the drawbacks of using Scrum-of-Scrums for scaling in large software development is that it is a technique and not a complete methodology. In many cases, it thereby ends up not being enough to scale Scrum successfully.

2.5 Scaling frameworks

Scrum-of-Scrums is one suggestion to scaling the agile methodology Scrum; however, in the recent years, several frameworks for scaling agile practices have also been introduced e.g. SAFe and LeSS which is two of the most well known. These frameworks bring additional practices for scaling agile practices (Dikert et al., 2016).

SAFe (Scaled Agile Framework):

SAFe is a framework which enables adoption of agile and lean practices to enterprise scale. The framework helps to synchronise alignment, collaboration, and delivery for a large number of agile teams [10].

There are three levels with SAFe; team, program, and portfolio. The **teams** are working agile with XP (Extreme Programming) methods. **Program** consist of 5-10 agile teams creating what is called an 'Agile Release Train' (ART) which is the primary value delivery in SAFe. ART consist of a value stream, or sequence of activities, which is supposed to produce a valuable set of

[8] https://www.agilest.org/scaled-agile/scrum-of-scrums/ *Accessed: 14.02.17*

[9] http://agileunleashed.blogspot.no/2009/08/scrum-of-scrums-approach-for-scaling.html *Accessed:15.02.17*

[10] scaledagileframework.com *Accessed: 01.06.17*

deliveries. **Portfolio** management requires strategy, investment funding, program management and governance SAFe is intended to allow large organisations to adapt to a more agile way of working. While agile methods tend to focus on the team level, SAFe represents a unified view of work for the whole organisation [11].

SAFe have been questioned a bit as it lacks empirical evidence. Some of the criticisms are that it is too top down, too many separate methods thrown together and a bit over simplified that one way fits all [12] [13].

LeSS (Large-Scale Scrum):

LeSS is a framework that extends Scrum with specific guidelines and rules for scaling that try to preserve the principles of Scrum when adopting it with multiple teams. It is important to notice that LeSS is not new and improved Scrum, nor Scrum at the bottom of each team [14]. However, it is how principles, purposes, and elements of Scrum can be applied in a large-scale context. LeSS is Scrum applied to many teams that often are either cross-functional or full-stack feature teams. On the contrary of Scrum that is used to one team that deliver one product, with LeSS the teams are working together delivering on a shared goal within a common sprint. The idea of LeSS is to have a framework that is minimalistic and allow the organisation to fill the framework instead of having too many rules, roles, artefacts, etc. within the framework [15].

While SAFe provides a way for large projects to organise themselves as teams of teams of agile teams, LeSS does somewhat the same; however, with more focus on how to improve communication between the teams [16]. These two frameworks have been significant contributions to understanding how to scale agile principles, and many organisations have incorporated one of these frameworks in their organisation. However, these frameworks offer limited, and often contradicting, solutions for scaling agile projects. The frameworks recognise for instance coordination and communication to be issues with large-scale agile; however, the solutions for solving them are often limited (Crowston et al., 2016).

[11] http://www.cio.com/article/2936942/enterprise-software/introducing-the-scaled-agile-framework.html *Accessed: 01.06.17*

[12] http://ronjeffries.com/xprog/articles/issues-with-safe/ *Accessed: 02.06.17*

[13] http://www.djaa.com/kanban-anti-safe-almost-decade-already *Accessed: 02.06.17*

[14] https://less.works/less/framework/introduction.html *Accessed: 02.06.17*

[15] https://www.agilealliance.org/resources/sessions/introduction-to-large-scale-scrum-less/ *Accessed: 02.06.17*

[16]http://www.cio.com/article/2974436/agile-development/comparing-scaling-agile-frameworks.html?page=2 *Accessed: 02.06.17*

Chapter 3

Coordination

This chapter will first be presenting an introduction to general coordination theory in Section 3.1, to give an insight of what is meant by coordination, and how different researchers have differences of opinion of what coordination means. After that, in Section 3.2, the coordination model Van de Ven will be presented to give an outline of the particular model which is applied in the analysis of this case study. Section 3.3, 3.4, 3.5 will give an deeper insight into the three different coordination modes of the Van de Ven model of coordination. Lastly, in Section 3.6, the hypotheses Van de Ven have defined will be presented.

3.1 Introduction to coordination

We all have some intuitive sense of what coordination means, and there exist several definitions of coordination. However, currently, it does not exist a single definition that is widely accepted by everyone. One of the early definitions was proposed by Malone (1988) where he defined coordination to be *'When multiple actors pursue goals together they have to do things to organise themselves that a single actor pursuing the same goals would not have to do. We call these extra organising activities coordination'.* Some years later, Malone and Crowston (1994) redefined that definition to be *'Coordination is the managing of dependencies'.* The definition is reasoned in that there are interdependent relationships between activities, and to cope with these relationships effectively, coordination mechanisms are needed (Deng et al., 2007).

Osifo (2012) refer to coordination to be classified as an element in an organisation. Further, he points out, what also can be found in another article (Bouckaert et al., 2006) that if there is no interdependence, then there is no need for coordination either, which substantiate Malone and Crowston's definition.
Coordination is necessary for the organisation both internally and externally. Internally, because coordination is crucial to accomplishing cooperation by having participation and transparency. If there is not internal coordination, adequate progress of the project will become difficult to achieve. Based on cooperation within the team, internal coordination also contributes to setting rules and standards. External coordination is also essential by defining boundaries to establish the right vision and focus for the project (Osifo, 2012).

Osifo (2012) summarizes the different aspects in a project where coordination is crucial.

> *'Coordination is a part of planning, because it tells what to include in a good plan and how to execute it. Coordination is part of organizing because it takes the first lead. Coordination is part of staffing, because it specifies who will be a staff and the rational placement. Coordination is part of directing, because it gives a clear focus. Coordination is coordinating. Coordination is a part of reporting, because it makes it realistic. Finally, coordination is part of budgeting, because it gives it a good appraisal'.*

As can be read in the quotation, coordination can be found in almost every part of a project, and should, therefore, be paid attention to be able to optimise the work. Coordination is seldom exercised alone through a single coordination mechanism. It is most often achieved through several mechanisms that all together achieve the overall coordination for a project or an organisation (Dietrich et al., 2013).

3.1.1 Coordination in large-scale projects

Software development in larger scale have in the later years met challenges, especially in relation to coordination between teams. As projects increase in size and complexity, the need for coordination also increases (Kraut and Streeter, 1995). There has been an increasing use of the team of team's setup in projects because of increasing complexity, and the organisations thereby need different approaches to handling these changes and challenges with coordination (Scheerer et al., 2014). When dealing with large groups of people that need to be coordinated, this often ends up in a hierarchical team of team's setup, which in organisational theory is defined as a multi-team system (MTS) (Scheerer et al., 2014). Mathieu et al.(2001) define MTS as:

> *'Two or more teams that interface directly and interdependently in response to environmental contingencies toward the accomplishment of collective goals. MTS boundaries are defined by virtue of the fact that all teams within the system, while pursuing different proximal goals, share at least one common distal goal; and in doing so exhibit input, process, and outcome interdependence with at least one other team in the system'.*

In software development, a common way of handling inter-team coordination, in a similar environment as MTS, is according to research done by Scheerer et al. (2014), the Scrum-of-Scrums approach that has been presented in Section 2.4, which is substantiated by other researchers as well (Larman, 2008, Schwaber, 2004).

When it comes to large software development projects, an increasing effort must be put into coordination to get all the work done and being able to work together without too much redundancy (Kraut and Streeter, 1995). In the process of developing software, there is a need for tight coordination among the involved to produce a successful system. Coordination is crucial in large projects, but can, however, be difficult to achieve.
Agile software development in large-scale deal with multiple teams that require some sort of inter-team coordination, because of increased complexity. When teams grow in size, the number of inter-team dependencies also tend to increase. More coordination effort is then needed to deal with inter-team dependencies so that each team's individual goal is reached, and also the overall goal of the project is achieved (Larman and Vodde, 2010, Paasivaara et al., 2012, Scheerer et al., 2014). Bick et al. (2016) have conducted a research where they studied five different ways of practising agile at scale. What they experienced in that study was that inter-team coordination approaches vary a lot when it comes to their nature of coordination (Bick et al., 2016).

Curtis et al.(1988) present in their study that coordination in large projects can be challenging, as there are more likely to be communication bottlenecks and breakdowns. Large-scale projects also come with more uncertainties that also affect coordination, as specifications of the requirements might change over time. This can be unpredictabilities when it comes to both the software, and the tasks the team members shall perform (Kraut and Streeter, 1995). There can also be uncertainties within the teams that affect coordination. For instance, what should be prioritised first and how to do tasks, as different people might have different opinions? If coordination between the various teams, and within the team, is weak, this alone can contribute to integration failure.

Marks et al. (2005) describe the boundary of a MTS based on Mathieu (2001) as when *'teams share input, process, and outcome interdependence with at least one other team in the MTS network'*. This has similarities to the case for agile teams in large-scale development projects, where the teams are dependent on the other teams to be able to deliver a product successfully. This substantiates the need for some kind of inter-team coordination. Another study that has pointed out inter-team coordination to be crucial is Melo et al. (2013). They identify agile team management to be the most influential factor to the productivity of agile teams. In relation to this, their case study showed that inter-team coordination emerged as an important inter-team management issue (Melo et al., 2013). Inter-team coordination influence team management productivity. If there, for instance, is a lack of commitment by one team, which can end up in delays and misalignment, or if there are too strict rules of coordination among the teams, this will lead to less agility.

Coordination in large-scale agile software development comes with several challenges that needs to be dealt with, and inter-team coordination is one of the challenges that demand extra attention to be able to succeed.

3.2 Van de Ven model of coordination

In organisational theory, Mintzberg (1980) proposes, based on earlier research, that mutual adjustment, direct supervision, and standardisation to be mechanisms for handling coordination. Strode et al. (2012) present valuable insight to coordination mechanisms related to agile development (in small-scale). They present a coordination strategy that includes the mechanisms synchronisation, structure, and boundary spanning to be valuable for effective coordination. Their study also points out that from an agile perspective, coordination is often achieved through mutual adjustment at a group level. At individual level, coordination is obtained by personal horizontal coordination by one-to-one communication (Strode et al., 2012).

Thompson (1967) divide coordination into three based on the type of inter-dependencies: pooled, sequential and reciprocal. Pooled are when units within an organisation accomplish completely separate tasks and do not interact implicit dependency to other entities. Standardisation best coordinates this with little communication and decision effort. Sequential arises when one unit depends on the output of another to continue its work. This is best coordinated by planning and medium effort for communication and decision effort. Reciprocal occurs when the input and output flow in both directions simultaneously between the dependent units and is coordinated by feedback and mutual adjustments.

Van de Ven et al. (1976) is another coordination model that is to some extent similar to the findings of the other theorists, but adds the dimension of *team* to their coordination strategy. This means, for instance, that mutual adjustment is extended by collective interactions within teams who are usually co-located. However, it has many similarities to how Thompson (1967) suggests in theory that coordination can be achieved. The Van de Ven model of coordination is, because of the team aspect, interesting concerning inter-team coordination, which has been mentioned to be an essential factor to coordination in large-scale projects.

According to research conducted by Van de Ven et al. (1976) there are three modes for coordinating work activities; *impersonal, personal, and group mode of coordination* which has several similarities to what is suggested by Thompson (1967). These will be presented in more depth in the next sections.

In the study Van de Ven et al. (1976) conducted, they wanted to examine to which extent the factors task uncertainty, task interdependence, and unit size could predict variations in the use of the three modes of coordination.
Van de Ven et al. (1976) state that there are some fundamental factors which explain why there are different mechanisms for coordination within an organisation; namely, task uncertainty, task interdependence, and unit size. They state that in different situations, it is possible to determine when one, or a combination of different coordination mechanisms, is used, dependent on these three factors.

Task uncertainty: is considered the difficulty and variability of the work. This can, for instance, be measured in how analysable the work is, and if the work methods are predictable (Van de Ven et al., 1976). Van de Ven et al. also list other measures of task uncertainty; 1) the degree of complexity of the search processes; 2) the amount of thinking time to solve problems; 3) the extent to which task processes or interventions have knowable outcomes; 4) the amount of time required before outcomes are known.

Task interdependence: concerns to what degree a task is dependent upon one another, and to what extent it is possible to do individual jobs separately.

Unit size: is in this context related to the total number of people employed in a work unit.

The three factors, which Van de Ven et al. (1976) mean are fundamental, are used to explain the usage of different coordination modes. The various modes are needed to be able to increase benefits of the project, as well as deal with potential challenges.

In the study conducted by Van de Ven (1976) it is also stated some interesting relations regarding what happens as unit size increases which is relevant for this research as it aim at examining agile methods in large-scale projects.

1. Group cohesiveness decreases and sub-group formation increases.

2. Member participation decreases and more mechanical methods are used to introduce information, and more direct attempt are made to control the behaviours of participants in reaching a solution.

3. Face-to-face techniques of leadership behaviour give away to more impersonal techniques of coordination.

4. Demands on the leaders become more complex and numerous, and group members become more tolerant of highly structured and directive leadership.

By these four aspects, Van de Van et al. identify the relation between size and team, which front that large-scale projects have influence on teams, and other coordination mechanisms are then needed.

Van de Ven's model of coordination has a high focus on the team aspect and how coordination modes change, depending on influential factors which may affect coordination mechanisms. Coordination of large-scale projects is complex, and from what can be learned from the Van de Ven model is that coordination is a changing mechanism, and it is rarely exercised through only one coordination mechanism. Several aspects influence coordination, and as projects become larger, it becomes more challenging to deal with it appropriately.

The next sections will present more in-depth the different modes of coordination; impersonal, personal, and group mode, and substantiate them with theory from other studies to see their relation to inter-team coordination. Zmud (1980) describes these modes to be three predominant coordination modes that enable information processing. Espinosa (2004) suggest in their study that different modes of coordination are needed as the coordination modes are suitable for various tasks. They also point out the fact that the same task might require different coordination modes over time (Espinosa et al., 2004). This substantiates Van de Ven's model of dividing coordination into coordination modes suited for different purposes. Van de Ven et al. (1976) suggest that coordination mechanisms within each of the different modes are used in various combinations to achieve a collective goal.

3.3 Impersonal mode of coordination

Impersonal mode of coordination relates to anything that has to do with programming, administrative coordination and technical tools, and once it is implemented, its use requires minimal verbal communication between actors (Boos et al., 2011, Van de Ven et al., 1976). These principles are also suggested in theory by Kraut et al. (1995), that point out the combination of large size, uncertainty, and interdependence require particular coordination techniques like technical tools, modularisation and formal procedures. These techniques do not remove all challenges of coordination but can help to ease some of them.

Table 3.1 state how this research have chosen to define the impersonal mode of coordination, together with some examples of coordination mechanisms to give a clearer overview of the meaning.

Coordination category	**Definition**	**Examples of coordination mechanisms**
Impersonal mode	Use of codified blueprint of action that is impersonally specified (e.g. use of plans, schedules, rules, and policies in coordination)	Functionality reports and documents in sharing data, use of common database, process documentation, implementation schedules, intranet pages, sales plans, IT tools

Table 3.1: Impersonal mode of coordination (Dietrich et al., 2013)

Impersonal mode of coordination helps to ease to coordination issues within a large project, and is often seen as increased in importance as a project gets more complex. Mintzberg (1980) conducted one of the earlier studies on coordination, and suggested several coordination mechanisms that could contribute to how organisations could coordinate their work more effective. One of these mechanisms, namely standardisation, substantiate the impersonal mode of coordination that Van de Ven et al. (1976) have identified. According to Mintzberg (1980) standardisation involve three aspects.

Standardisation of skills: are often the case in the initial phase of a project for instance, where coordination is achieved through standardisation of skills and knowledge through training and education.

Standardisation of work processes: where one are using standards, e.g. rules, routines or regulations, to guide how to perform a certain activity.

Standardisation of outputs: gives coordination by communicating and clarifying what is expected of the results.

Another study which also has similarities to impersonal mode of coordination is one of the classifications of Espinosa's coordination mechanisms (Espinosa et al., 2010). Espinosa (2010) define mechanistic coordination to concern coordination by a program or by a plan with the use of artefacts, processes, and routines to deal with dependencies with little communication. The definition of mechanistic coordination is one of three mechanisms Espinosa (2010) identify to deal with coordination, and as Van de Ven et al. (1976), Espinosa state that there is a need for several mechanisms to deal with coordination.

As can be found in Table 3.1, the use of tools to organise and have a common platform to share reports, guidelines, etc. are important in large-scale projects to keep coordinated. Malone and Crowston (1994) contributed, with their early work on coordination, with a theoretical modelling framework to be used for analysing complex coordination processes. They saw the benefit of using this framework to examine group action, regarding actors performing interdependent tasks, which either create or require different resources (Crowston et al., 2004). Tasks can be seen as system requirements that are translated from the requirements of the customer. While resources contain information like problem description, existing system functionality, and time and cost analyses (Crowston et al., 2004). However, the theory has shown in more recent times that it is not that suitable to predict for instance coordination effectiveness and that not all coordination mechanisms can be seen as general. That said, it is still a valuable framework to gain a better understanding of how the different factors support coordination (Strode et al., 2012). This also substantiates that some frameworks, guidelines, plans - some artefacts to keep organised and coordinated - are necessary, as with impersonal mode of coordination.

3.4 Personal mode of coordination

With the personal mode of coordination, Van de Ven et al. (1976) identify that coordination occurs as feedback by mutual adjustments regarding the input information one receive. The individual makes mutual task adjustments either through vertical or horizontal communication channels. Vertical communication is usually line managers and unit supervisors, while horizontal communication is concerned about individuals in teams having one-to-one communication with someone. With horizontal communication, there are often a non-hierarchical relationship between the actors.
As suggested in theory by Mintzberg (1980), there are two coordination mechanisms, namely mutual adjustment and direct supervision which concerns the same principles as with vertical and horizontal communication in personal mode of coordination.

Mutual adjustment: are when team members are using informal communication with others to coordinate their interdependent work.

Direct supervision: where there typically are one individual, e.g. the manager, that gives specific orders to the rest of the team, and the team thereby coordinate and take responsibility for that work is being conducted.

These two mechanisms substantiate the need for personal mode of coordination, and that through mutual adjustments within a team, coordination can be achieved. Also, Espinosa's (2010) classification of organic coordination has some similar aspects as Mintzberg in relation to personal mode of coordination. Organic coordination is related to coordination that is achieved by feedback or by mutual adjustment, and thereby the coordination is mainly accomplished by communication and interaction. The communication can be informal and spontaneous or formal and planned (Espinosa et al., 2010).

Table 3.2 state how this research have chosen to define personal mode of coordination, together with some examples of coordination mechanisms to give a clearer overview of the meaning.

Coordination category	**Definition**	**Examples of coordination mechanisms**
Personal mode	Use of mechanisms in which individual role occupants make mutual task adjustments through vertical or horizontal communication	Direct face-to-face contacts, phone calls, use of external consultant as liaison, use of same resources in several teams, team managers participation in other teams' meetings, project manager's participation in the work of several teams.

Table 3.2: Personal mode of coordination (Dietrich et al., 2013)

Personal mode of coordination includes both formal and informal communication which have been identified to be important to coordination (Kraut and Streeter, 1995). Especially, when dealing with a high degree of uncertainty, Kraut et al. (1995) state that the informal, interpersonal communication is valuable both for team members and for the project as a whole. Formal and informal communication bring value to a project, as coordination increases when sharing information. Another important factor for coordination in software development, as Kraut et al. (1995) point out, is that the personal communication that finds place across functional boundaries that help to handle uncertainty. This substantiate that personal mode of coordination is a critical mode to deal with inter-team coordination to ease uncertainty.

Theory by Boos et al. (2011) describe personal mode as a useful coordination mode when things are not scheduled and anticipated, and they put communication between the team members, as a dependency factor for the personal mode of coordination. This can also be found in theory by Dickinson and McIntyre (1997) which identify communication as "the glue" of teamwork since it links all the other components. Boos et al. (2011) identify several measurement levels for whether personal mode will succeed, such as planning, information exchange, feedback; by giving, seeking and receiving information between teams, and leadership (Boos et al., 2011). As suggested in theory by Kraut et al. (1995), coordination via communication does not only occur formally via meetings and documents, on the contrary, a substantial amount of coordination happens with informal communication for instance in cafeterias or hallways. This substantiate the advantages of personal mode of coordination

As personal mode of coordination mainly concerns communication, both horizontally and vertically, some factors challenge this coordination mode which can be crucial to whether the coordination is successful or not. Lehtimäki (1996) state the importance trust has with coordination. This is caused by coordination creates the network where organisational performance is understood. Trust is then vital when project increase in complexity, but may also be harder to accomplish then. Moore et al. (2008) state out that to succeed with agile methods one must trust decisions that are made without your impact of control, and that this trust must also be extended to the whole project. Better performance can be achieved by having good coordination, and then the network of trust is essential for coordination (Osifo, 2012). The lack of trust has been identified as a central factor to poor coordination and cooperation (Smith and Schwegler, Smith and Schwegler). Osifo (2012) concludes, based on various coordination literature, that it is visible that trust is a part of performance since it creates a foundation for proper coordination.

3.5 Group mode of coordination

With group mode, coordination also occur by feedback as mutual adjustments as with personal mode of coordination; however, the mutual adjustments occur through the group by scheduled or unscheduled staff or committee meetings Van de Ven et al. (1976). It involves new routine and is usually more planned communication.

Table 3.3 state how this research have chosen to define group mode of coordination, together with some examples of coordination mechanisms to give a clearer overview of the meaning.

Coordination category	**Definition**	**Examples of coordination mechanisms**
Group mode	Use of mechanisms in which individual adjustments occur in group of occupants (more than two) through meetings	Weekly status review meetings, informal inter-team meetings, delivery approval workshops, coordination group meetings, kick-off meeting, brown bag lunch meetings

Table 3.3: Group of coordination (Dietrich et al., 2013)

Group mode of coordination is an important mode, especially concerning large-scale projects where several teams need to be coordinated. Espinosa's (2010) classification of cognitive coordination refers to the knowledge the actors have about each other, as well as the tasks the others are doing. This coordination can be beneficial in relation to knowing what others are likely to do and with group mode one can achieve coordination by having meetings evenly to keep updated on each other. Espinosa et al. (2010) state that cognitive coordination can be seen as a critical enhancer of mechanistic and organic coordination which is important for impersonal and personal mode of coordination.

Other researchers have also identified aspects of group mode of coordination to be important to deal with coordination in large projects. Dietrich et al. (2013) identified three specific coordination mechanisms through their study of multi-team projects which are centralised, decentralised, and balanced patterns. *Centralised coordination* concerns coordination that occurs at group level, like scheduled and unscheduled meetings to make adjustments.*Decentralised coordination* concerns coordination that occurs between team members, which not necessarily are pre-defined meetings. *Balanced coordination* involves a combination of the two previous.
The results of their study show that the diversity in coordination practices have an influence on several aspects of the project like *'information sharing, workflow fluency between teams, the efficiency of the project, and learning outcomes'* (Dietrich et al., 2013). Dietrich's study substantiates the importance of having different kinds of coordination mechanisms with multi-team systems (Mathieu et al., 2001) to handle inter-team coordination, and that several levels of coordination are important to consider when many teams are involved. Thereby, group mode of coordination is considered essential in large-scale projects.

In large-scale projects, one can find several kinds of dependencies that urge the need of coordination. With the proper coordination, it enables collaboration among the different teams (Melo et al., 2013). Dependencies need to be dealt with and compatible with the needs of the teams. Malone and Crowston (1994) conducted an interdisciplinary study of coordination where the key insight was that coordination could be seen as the process of managing dependencies among activities. These activities can be seen as constraints on an action in a situation (Malone and Crowston, 1994, Strode et al., 2012). A result of their study is their definition of coordination that is *'managing dependencies between activities'*. Their theory is built on ideas from the fields of organisation theory, economics, management, and computer science, and can be thereby being seen as an interdisciplinary study of coordination. Group mode of coordination is one way of dealing with this as dependencies will always occur, and these needs to be addressed also across the teams. That mode of coordination is suited to be able those dependencies across the teams, and is thereby necessary in large-scale agile.

Strode et al. (2012) have developed a coordination strategy that includes several mechanisms to cope with dependencies in a situation. By identifying several specific strategies for unique cases, the formation of general coordination strategy concept was developed (Strode et al., 2012). The coordination strategy concept then resulted in three main components: synchronisation, structure, and boundary spanning which are mechanisms to help manage dependencies. Synchronisation is of particular importance concerning group mode of coordination. It is achieved through synchronisation activities and artefacts that are produced during those activities. These activities are meant to bring the whole team together at the same time and place for a pre-arranged purpose. Many of these activities often occur only once, for instance at the beginning of a project to agree upon technical decisions, to develop a high-level project scope, and to define the initial requirements. However, there are also synchronisation activities during the project. The purpose of these activities is to gain a common understanding which is really important across the teams and thereby important for group mode.

The coordination that occurs with group mode is often directly related to inter-team coordination and issues that are necessary to deal with across the teams. This mode is important to have a balanced coordination (Dietrich et al., 2013) to increase information sharing, efficiency, workflow fluency and overall result of a project.

3.6 Van de Ven hypotheses

Based on the three modes of coordination that is presented in this chapter, Van de Ven et al. (1976) have identified three hypotheses based on the factors that are identified in Section 3.2, namely task uncertainty, task interdependence and unit size. One hypothesis for each of the factors which will be presented next.

Hypothesis 1: Increases in the degree of task uncertainty for an organizational unit associated with:

1. a lower use of the impersonal coordination mode
2. a greater use of the personal coordination mode
3. a significantly greater use of the group coordination mode

Hypothesis 2: Increases in workflow interdependence from independent to sequential to reciprocal to teams arrangements will be associated with:

1. small increases in the use of impersonal coordination mechanisms
2. moderate increases in use of personal coordination mechanisms
3. large increases in use of group coordination mechanisms

Hypothesis 3: An increase in work unit size is associated with:

1. a decrease in use of group coordination
2. an increase in the use of personal coordination
3. a significant increase in use of impersonal coordination mechanisms

By looking at these hypotheses identified by Van de Ven, one can observe that the connection between unit size and the three modes of coordination is opposite of what can be found regarding task uncertainty and task interdependence. With more task uncertainty and task interdependence, there are more likely to be an increasing usage of group and personal mode of coordination. The opposite scenario is when a work unit size increases, impersonal mode of coordination will increase, while the two others decrease a bit.

Part III

Research methodologies

Chapter 4

Method

This chapter will present the research methods that are the foundation for this master thesis. First, in Section 4.1 a brief description of the literature review that was conducted in the preliminary project will be presented. Then, in Section 4.2 the research method and approach for this study will be elaborated, including case selection, how the data was collected and how the analysis was conducted. Last, in Section 4.3 the quality of the conducted case study is evaluated.

4.1 Literature review

The literature review that was conducted before the master thesis has been an important part of the research (Østdahl, 2016). With carrying out a literature review, it allowed exploring the existing literature regarding large-scale agile software development. The aim of the literature review was to discover, analyse, and evaluate what has been studied before, and sustain my research and place it in context.

The topic of this master thesis somewhat differs from the literature review which was conducted Fall 2016; however, both are within the area large-scale agile software development. Based on the results and insight gained from the literature review an emerging topic caught interest, which could be interesting to study further, namely inter-team coordination. Experiences from the literature review was that inter-team coordination was pointed out to be an essential factor to succeed with agile software development in large-scale, but there was also a gap in the literature on successful empirical studies within this area.

Based on the new perspective on large-scale agile software development, new searches were conducted to get a broader intuition on what had been researched earlier regarding inter-team coordination. Also, in this phase, the research aimed to look at what insight that could be gained from general coordination theory.

A new shorter literature review was conducted for large-scale agile software development, but this time the focus was turned to inter-team coordination, and also coordination theory in general. The main searches were done through the databases found in Table 4.1. All these databases are useful platforms to find relevant and trustworthy articles.

ISI Web of Science	apps.webofknowledge.com
Science Direct - Elsevier	sciencedirect.com
ACM Digital Library	dl.acm.org
Google Scholar	scholar.google.com

Table 4.1: Databases for literature search

Several searches were done to find the best and relevant articles. The search for inter-team coordination was also restricted the search results in Web of Science to the categories *Computer Science Software Engineering* and *Computer Science Information System* since those categories were most related to software development which is the focus of research.

Many references of the articles were further checked closer which is called snowball sampling [1]. This is a way of finding more recognised papers that could be relevant to read closer. Several articles, which were useful to the research, were found this way.

Related to the topic agile software development and the subject coordination there are many articles so it was important to find articles with high credibility and those that could be related to the research questions that can be found in Section 1.2. Articles that were thoroughly and objectively conducted through extensive empirical study were chosen over lessons learned articles from one person's experience. This narrowed the results of the searches, but again the articles were carefully chosen over others because of their relevance and thorough work of the researchers. By doing a search through scientific databases, which are widely used in the research community, instead of the world wide web, one also ensures getting good and relevant articles faster. Also, one can easily find articles that are cited by other researchers which can be very useful.

4.2 Research method

Through conducting a literature review, it was evident that there is a gap in research studies regarding inter-team coordination in agile software development in large-scale. Thereby an exploratory case study was chosen for further research. An exploratory study was suitable as there is a lack of literature about the topic, and by focusing on the depth of a real-life project, one can identify topics that can be examined further (Oates, 2006).

The intention of this exploratory study is not to give final or conclusive solutions, on the contrary, the purpose is to help give a better understanding of inter-team coordination. An ambition is to be able to explore the topic inter-team coordination at varying levels of depth. One can recognise an exploratory study by exploring topics. Some of the advantages of conducting an exploratory study are that it is flexible and adaptable to change, and also suitable for laying groundwork which may be potential future studies. One of the disadvantages of this kind of study is that it will only cover one case which is explored, meaning the results may not sufficiently represent the reality for all types of cases of large-scale agile development. This has similarities to interpretive research as it seeks to understand and explore how factors of inter-team coordination are in a particular social setting. Interpretivism is recognised by factors like multiple subjective realities, study of people in their natural social setting, multiple interpretations and qualitative data analysis (Oates, 2006). These are all factors that are relevant for this exploratory case study.

[1] http://research-methodology.net/sampling-in-primary-data-collection/snowball-sampling/ *Accessed:25.09.2016*

4.2.1 Case selection

Interview material was given, in collaboration with Agile 2.0, as they were interested in having new insight in the material they had collected from focus groups with persons who had been involved in a very-large scale agile development project. The examined case is said to be one of the first large-scale software development projects in Norway and lasted from 2008 to 2012. The purpose of the project, from now called Omega were to create a new office automation system. During the Omega project, there were at most 13 agile development teams involved. There were six Gamma-teams, and also four Alfa-teams and three Beta-teams that were hired as contracted consulting companies in Omega. This case is, therefore, a very well suited to explore in-depth regarding inter-team coordination.

4.2.2 Data collection

Focus groups were used to collect data in the exploratory case study. Using focus groups are a preferred way to accumulate a considerable amount of useful data. There were in total conducted three focus groups, one for each of the involved organisations (Gamma, Alfa, and Beta). The topic of the focus groups was to examine *inter-team coordination* and *knowledge sharing*.

By having a focus group for each of the organisations, it was more likely to achieve openness, and easier to gather data concerning particular organisation without losing too much valuable data. Focus groups take advantage of communication between the participants to generate data. These kind of group interviews are beneficial to gain information from several people simultaneously, which are both efficient but also gives the benefit of group interaction which is part of the method. This interaction is often valuable, and individuals in focus groups are often encouraged to talk and interact with each other freely.

There are often differences in culture and working habits between different organisations. If the focus groups were held with all the three organisations together, it could have affected the result. The composition of the focus group can influence the result, and the data can, for instance, be affected by hierarchy within the group if people are not willing to be open (Kitzinger, 2005).

There were several roles represented from the Omega project; project leaders, scrum-masters, and functional and technical architects. There was a total of eight participants in the focus groups, and in Table 4.2, the different representatives from the organisations are summarised.

Organization	**Number of participants**	**Roles**
Alpha	2	Project leaders
Beta	3	Technical architect, Project leader, Functional architect
Gamma	3	Project leader, Scrum master, Technical architect

Table 4.2: Participants in focus groups

4.2.3 Data analysis

To be able to see patterns and similarities of the three focus groups, an extensive thematic analysis was required. There were a lot of qualitative data to go through, and it was necessary to analyse it to abstract the themes and patterns that could be most relevant in relation to the research questions. Thematic analysis was used to analyse the textual data.
Thematic analysis is a preferred method to gain information and try to make sense of unrelated data

[2].It is a systematical approach that is used to analyse the qualitative information and gain knowledge and was suitable to analyse the transcriptions from the focus groups. Thematic analysis gives the researcher the opportunity gain a deeper understanding and insight of the area they are researching from the collected data.

Once the data is collected from the sources, in this case through focus groups, the next step in a thematic analysis is to code the data. This can be done by hand or through a computer-aided analysing tool, as was done here with NVivo [3].

NVivo

The gathered data consisted of three reports, one for each focus group. The first step was to read through them two times to get a better overview of the material. Several computer-aided analysing tools were considered to start analysing the content, but I choose to use NVivo because it seemed sturdy and that it had many opportunities within one tool.

NVivo is a tool for analysing unstructured material that is valuable for qualitative research. Qualitative research is beneficial to gain a better understanding of underlying reasons, motivation and different opinions. It is suitable to use when you are dealing with rich text-based data that require an in-depth level of analysis. NVivo makes it easier to classify, sort and organise the collected information by coding the text into nodes and cases. This type of coding enables opportunities as exploring relationships in the data and then combine analysis to create links, shape the data, and create models. With NVivo it is easier to cross-examine the information in several ways like a search engine and query functions.

Research model

For further examination and to be able to structure and organise the data it was coded into nodes. To be able to categorise the data I choose to frame the analysis by using a coordination model; namely, Van de Ven model of coordination which is described in Section 3.2. This model was chosen as it combines several other well-known coordination theories and because *team* is a crucial factor in that model which is relevant for inter-team coordination.

The researcher have not identified other studies that use the Van de Ven model of coordination to examine inter-team coordination in large-scale agile development (with exception of a newly released paper (Dingsøyr et al., 2017)), but it has been applied to study coordination for other purposes earlier (Dietrich et al., 2013, Zmud, 1980).

By using a particular model for framing the analysis, one can gain valuable insight based on factors that researchers have identified to be important. The in-depth analysis will then either confirm Van de Ven's division of coordination and influential factors, which give more grounding for that model. Or, the results of the analysis will identify vulnerabilities of using that model for gaining insight concerning inter-team coordination. Through the thematic analysis, the coded items were recognised by factors of the Van de Ven model and were conducted in several rounds as will be described next.

Coding analysis

The first rounds of coding focused on the three modes of coordination as Ven de Ven identifies namely, group, personal, and impersonal mode of coordination. Definition and examples of the three coordination modes can be found in Table 3.1, 3.2 and 3.3 in Section 3.3-3.5. It was chosen to code it at that level first to get an even better insight of the collected data and gain a better

[2] http://designresearchtechniques.com/casestudies/thematic-analysis/ *Accessed: 01-06-2017*

[3] http://www.qsrinternational.com/nvivo-product *Accessed:20-05-2017*

understanding of the model. The analysis had a primary focus on aspects of that model, and in the first rounds of coding other factors, that might be important to the research questions, were excluded from conducting the analysis according to the chosen research model.
Further, Van de Ven has identified three factors that influence coordination, which are: task uncertainty, task interdependence and unit size (Section 3.2). These factors were then used in the next round of coding to get insight at a deeper level of the data. When these had been coded at nodes as well, it was possible to see relations between different modes and the factors.
Lastly, there was a round of coding, where relevant findings regarding inter-team coordination as well as agile practices, were coded. This was done to be able to see whether there were some specific quotes for these factors that were worth noticing and if there were any connections to the different nodes coded with Van de Ven model.

In a thematic analysis, the code should be validated and reviewed by more than one person to ensure integrity, so the material has not been misinterpreted and is free from researcher bias [4].This was challenging as this research is conducted by one person; however, some uncertainties around coding were asked the supervisor for clarification. The researcher did re-read the data and double-checked (and triple-check) the coded nodes to validate and make sure the coding were done as consistent as possible.

The second phase of the analysis consisted of comparing the coded nodes, by cross-checking different observed patterns, by for instance looking at frequency of coded mentions. This phase consisted of trying to run different queries, and text searches, to see connections and if there were any patterns between the various modes, in comparison with the different factors Van de Ven mention. By running different queries in NVivo, one could gain a deeper understanding and insight of how inter-team coordination was used and achieved in Omega.

4.3 Quality of case study

One of the weaknesses of this conducted analysis is that it was only analysed by one researcher. Runeson & Höst (2009) state that an analysis benefit from being carried out by many researchers, as that help to reduce bias from one individual researcher. If the case had been analysed my several researchers together, the validity or trustworthiness of the results might have been higher. The reliability of the results of this case analysis is thereby a bit weak as it is very dependent on the point of view the researcher had. Hypothetically, another researcher should have been able to gain the same results by conducting the same study later (Runeson and Höst, 2009). This is unlikely to be exact the same; however, as a certain model of coordination was used in the analysis, the results are more likely to be somewhat similar at least. On the other hand, if there are different interpretations of the model and what the various factors consist of, as well as a different focus of what is important during coding, are all a threat to the validity of the results as it is influenced by the researcher's point of view.
Klein & Myers (1999) also identify several principles to evaluate the quality of the case study. They also bring up multiple interpretations and the possibility of biases to be essential factors when assessing the quality. Both interview objects and researchers may perceive situations differently which can have influence on the results.

The analysis quality of the case study that has been conducted is not completely objective as

[4] http://designresearchtechniques.com/casestudies/thematic-analysis/ *Accessed: 01-06-2017*

there will be some bias from the researcher. Although the results of the analysis have not been influenced by interactions with the interview objects, and thereby persuaded by their personalities and meanings, the observations are most likely not entirely independent from other researchers. Klein & Myers (1999) identify the interaction between the researcher and the subject to be an influential factor, and that substantiate the reliability of the results. The reliability of the results should be seen in the light that the exact same observations and situations are unlikely to be met in a repeat study, as different cases include different people and interactions which will most likely lead to some differences.

Runeson & Höst (2009) also mention that internal validity of the study should be considered, which concern whether one factor affects an investigated factor there is a risk that the investigated factor is also affected by a third factor. The internal validity of the conducted analysis is only seen in the light of the chosen research model; however, there can be other explanations or factors that will lead to the same results. If there are influential factors of different degree that are not identified through the analysis, this can be a threat to the internal validity. Threats to validity are however discussed to some extent in Section 6.6 where limitations of the results are presented. The chain of evidence from the data to the research questions and existing theory are presented quite well in the analysis as Runeson & Höst point out to be important; however, alternative perspectives and explanations are not that evident through the analysis as they could have been.

Another principle Klein & Myers (1999) mention is abstraction and generalisation. The results and conclusion of this case study have been tried to be substantiated both by what can be experienced in the analysed case as well as what the theory states. This is to verify the theoretical insight that is gained from the case study. However, this could maybe have been done even more thoroughly if it was not due to time limitations.

In the next chapter, there will be a presentation of the finalised information from the coded material. The transcriptions of the focus groups were conducted in Norwegian, and are thereby translated by the researcher so it could be presented in this report. The following quotations in Chapter 5, from interviewed persons, are translated freely by the researcher to be as similar as the original.

Part IV

Results and evaluation

Chapter 5

Results

In this chapter, a description of the very-large scale development project Omega will first be presented in Section 5.1. This will give a better overview of how the project was structured and conducted. Then follows the result of the analysis of the case based on the Van de Ven model, Section 3.2, which is performed with the analysing tool NVivo, Section 4.2.3.

In Section 5.2, a general introduction to the analysis of the Van de Ven model will be presented before the concrete results of the three different coordination modes will be found in Section 5.3, 5.4 and 5.5. These sections will first present findings directly associated with the particular coordination mode, then the three influential factors Van de Ven identify (Section 3.2) will be evaluated within each of the coordination modes.

5.1 The Omega Project

The Omega project started in January 2008 and lasted until March 2012, and it is considered to be one of the first large-scale agile IT projects that have been conducted in Norway. The purpose of the Omega project was to create a new office automation system, and it was carried out by the public organisation *Gamma*. The project was divided into 12 releases, with a total of 2,500 user-stories. In total, there were 175 people involved, and of these, 100 were external consultants from five companies (where *Alfa* and *Beta* were the main external companies).

The reason to conduct this project was due to a public reform that required new functionality in office automation. It was chosen to use agile development principles for the project, as the public reform was not known at the beginning of the project.

In the preliminary project to this master thesis, conducted by the writing researcher, different organisations structures were listed (Østdahl, 2016). Matrix structure was then mentioned to be a suited organisation structure for agile development. That is also the structure that has been used in the Omega project which had several intersecting projects. The advantage of using matrix structure is that it is a hybrid of the divisional and functional structure with dual management. The information exchange is known to be efficient and close across the different teams. It also enables decision-making at the team level, and the team members feel as if they can contribute to the progress.

Omega was organised by having a program director at the top who focused mainly on external relations, and a program manager who focused on operations. Then, there were four project managers below them who was responsible for the different areas of the department; business, construction, architecture, and testing. A description of the work tasks of the various departments is presented below.

- *Business* - Responsible for analysis of needs through defining and prioritising epics and user stories in a product backlog.
- *Architecture* - Responsible for defining the overall architecture in the programme and for detailing user stories in the solution description phase. Consisted of a lead architect and technical architects from the feature teams. Suppliers from Alfa and Beta participated on a time and material basis
- *Development* - Divided into three sub-projects: one led by the public department Gamma (6 teams) with their own people and people from five consulting companies, and the two other sub-projects led by Alfa (4 teams) and Beta (3 teams). The feature teams worked according to Scrum with three-week iterations, delivering on a common demonstration day.
- *Test* - Responsible for testing procedures and approving deliveries from the development teams.

A visualisation of the division of Omega can be seen in Figure 5.1, where it is visualised how the Omega project was divided into teams between the main contractor Gamma, and the external consultants from Alfa and Gamma. There were an increasing number of teams throughout the project, and at the most, there were 13 agile feature teams involved.

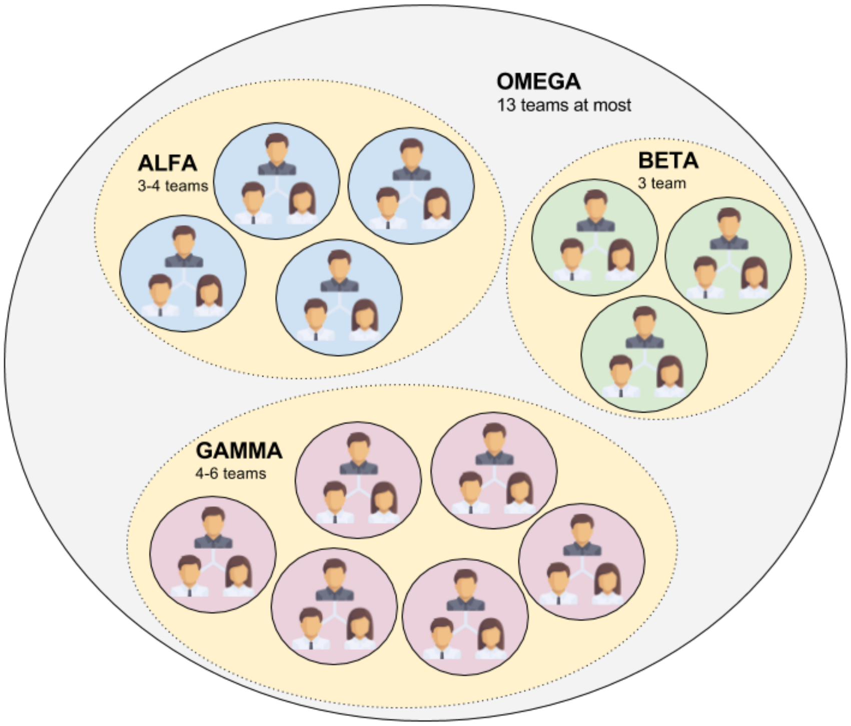

Figure 5.1: Division of teams in Omega project

The first year of the project all the teams were not co-located; however, from 2009 all the teams were moved to another office building, which enabled the teams could work in the same open space. Co-location has proven to be a major factor, and Figure 5.2 shows how the teams were organised in the open area when there were 11 development teams.

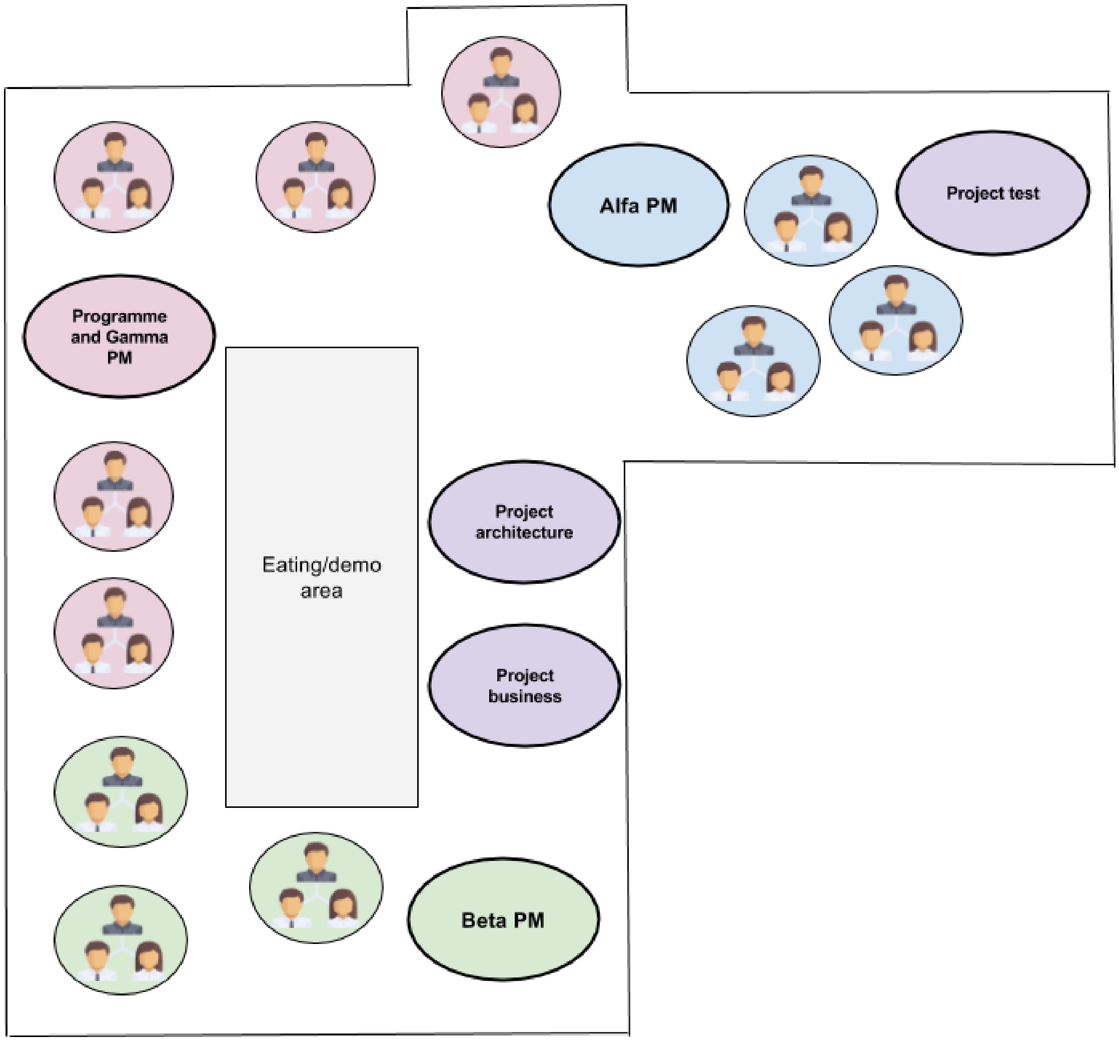

Figure 5.2: Location of teams

Being co-located brings both challenges and advantages in a very-large scale agile project. An advantage is that it is easy access to knowledge, meaning you most likely have all the needed competence within the same location. That can be beneficial regarding clarifications, and getting the best help. Another huge advantage is the informal communication you gain by being co-located. This has proven to bring many conveniences to agile projects, especially concerning coordination. Some of the drawbacks, when such a big project is co-located are that you need quite a big location; otherwise, it can fast feel a bit crowded and noisy. Also, with everyone in the same place, there are many people to relate to, which can be both positive and negative.

Scrum (Section 2.2) was the primary methodology that was used throughout this project. The development teams consisted mainly of eight to nine members. The roles of the different team members can be found in Table 5.1 (Dingsøyr et al., 2017). Although there were roles, the team members were each described as cross-functional. Two examples mentioned in the focus groups of this are: a tester was, for instance, 60% tester, but also 30% developer, and 10% designer. Same with a scrum master was 50% leader, and also 30% architect, and 20% developer. A result of this is highly competent people who were able to work in several fields, and was not restrained by competence. The result of this is not only individuals being cross-functional but also the teams and the whole organisation. This is mentioned in Section 2.1 to be beneficial for agile development.

Role	**Description**
Scrum master	Facilitated daily meetings, iteration planning, demonstration and retrospective
Functional architect	Responsible for detailing of needs. This role was usually allocated 50% to analysis and design, and 50% to development. Participated in project business
Technical architect	Responsible for technical design, working 50% on this and 50% on development. Participated in project architecture.
Test responsible	Made sure testing was conducted at team level (unit, integration, system and system integration testing). Delivered test criteria to the project test
Developers	4-5 developers were allocated to a team (both junior and senior developers)

Table 5.1: Roles in scrum teams (Dingsøyr et al., 2017)

Scrum was chosen mainly because of uncertainties regarding how the resulting system would be; thereby Scrum was suitable as it can easily deal with changing conditions (Section 2.2). The development teams worked in three weeks sprints with a joint demonstration day at the end which kept the teams up-to-date on each other.

Many team members had never used Scrum as a methodology before, and several team members felt extra motivated to conduct an agile project to gain a lot of new experience.

> *'Everyone in Gamma were excited to carry out this agile, and we wanted to succeed, and then you were willing to do everything to make it work. We did quite significant changes during the first year to make it work ... People were eager to read literature, go to agile conferences and gain as much experience as possible, everybody did it, and came back with new ideas and thoughts. People were willing to try new things where they saw things did not work that well'.* - Project Manager, Alfa

There were an eagerness to learn how to use agile methodologies in the Omega project, which of course influence the motivation of the team members. Team members might also affect each other positively since being around enthusiastic and motivated people are a motivation itself.

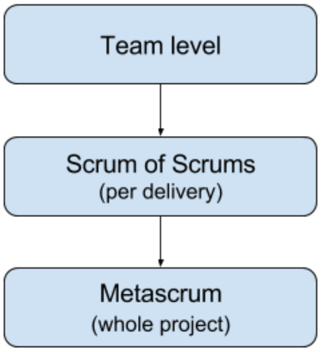

Figure 5.3: Hierarchy of scrum meetings

There was a division of the scrum meetings into a hierarchy, see Figure 5.3. There were daily scrum meetings at feature team level. In Section 2.2, scrum meetings are listed as an important artefact for the methodology. In Omega there were at first daily Scrum-of-Scrums meetings (Section 2.4), were the scrum master from each team were represented. However, later it was reduced to being three times a week instead. The Scrum-of-Scrums meetings were within the contracting teams. Then, twice a week there was meta scrum[1] across the whole project, were the attendants were the project leaders of the different contractors, as well as the leaders for the various sub-projects like business, testing and architecture. This division is similar to the division of Scrum-of-Scrums and Scrum-of-Scrums-of-Scrums, which can be found in Section 2.4 with Figure 2.2.

5.2 Van de Ven model

The findings in this section will be based on the Omega project which is presented in the previous section, Section 5.1.

The Van de Ven model (Section 4.2.3) was chosen as the research model for this analysis. The findings will, therefore, be based on the factors that the model consists of, and the results will thereby be drawn from what can be learned by using that particular model.
The Van de Ven model put a high emphasis on coordination with a focus on teams (Section 3.2) and was, therefore, a suitable and interesting model to use to analyse inter-team coordination in this case.

Through the analysis, it was remarkable that several arenas and mechanisms were used for coordination and communication. These arenas also changed over time, as the informal communication increased, which was seen valuable. That was a result of increased trust between the team members and across the teams.

Through analysing the transcriptions of the focus groups, several coordination arenas and methods have been mentioned, which have been used partially or throughout the whole project time. In Table 5.2 some of those coordination mechanisms, which were referred to in the focus groups, are listed. There were more mechanisms used for coordination; however, these stood most out. This table is included to give a quick overview over the most important arenas for coordination, as well as showing how many arenas and methods of coordination that have been used in the project. This indicates

[1] Meta Scrum is by some defined to be the same as Scrum-of-Scrums, but in this case it is the highest level meeting which can remind of what some call Scrum-of-Scrums-of-Scrums

that inter-team coordination fast can become complex. These will be presented closer in relation to the coordination modes in the next sections.

Arena or method of coordination	**Description**
Stand-up meetings	Was used daily within the different Scrum teams. Short status update meetings, often around the team board where challenges and progress was discussed orally. Tried to keep it to maximum 15 min
Scrum of Scrums	These meetings were used, first every day, then later two-three times a week. They were held within the different contracting teams (Alfa, Beta and Gamma). In the Scrum-of-Scrums meetings the scrum master from each scrum team were represented, as well as representatives from the project managers. The purpose of Scrum-of-Scrums were to identify and handle obstacles.
Meta scrum	A meeting that is somewhat similar to Scrum-of-Scrums but with less details which was held twice a week. Represented at the meta scrum meetings were the project leaders, and the sub-project leaders from business, architecture, development and testing
Planning day	This was used in Omega as kind of a kick-off for each sprint. The project owner and all the team members were present. On these day the following sprint iteration was planned and arena of focus was decided at project, organization (Alfa, Beta and Gamma), and team level.
Retrospective	Were used both within the Scrum teams as well as in both solution description team and project management team. Within the Scrum teams it was one a week - every Friday. These meetings were documented in Confluence.
Demo	A demo was held as a presentation for each Scrum team after each sprint were anyone who had interest in the work could attend. It was tried to keep it at a maximum of 10 min for each team. There were also a larger demonstration with the product owner when a release was finished.
Solution description	Was a meeting where a master plan for the whole project was developed at the beginning of the project, and it was continuously improved throughout the project. In the solution description meetings aspects like coordination and dependencies across the organization were discussed.
Dependency meeting	This meeting was held with all the scrum masters from the three organizations. The purpose of this meetings were to discuss what the teams planned to do to easier notice dependencies that might be across the teams, which might needed to be considered. This meeting was conducted on the planning day.

Table 5.2: Coordination arenas and methods

Front-end meeting	Was an own meeting for the front-end developers, as coordination between the teams working on front-end was needed and to share experiences with the framework that was used
Architecture forum	Was introduced a bit out in the project; however, it was very valuable. It was conducted once a week across the teams, where each team's architect presented the tasks for their iteration.
Open-space	Were tested to use in Omega, but was dismissed after 3-4 times. It is an arena where people can suggest topics to discuss, and it based on exchanging knowledge, experiences and have discussions on a voluntary basis, or after need.
Board	The whiteboard was an important artefact to coordination, and was used for discussions and status updates. The stand-up meetings were around the boards, and was a visualised artefact for the current sprint.
Bug-board	This was used as a discussion arena by the quality assurance team were the testers had meetings around the boards, more frequently around releases and acceptance testing. The purpose of this was to identify bugs, and allocate it to the responsible Scrum team.
Collective coffee break	At 2PM every day there was a joint coffee break which also purposed as an arena for informal communication which is contributing to coordination across the teams as it often was small discussions and updates while drinking coffee
Jira/Wiki/Confluence	Different software and tools were used to keep track of the process, and all the documentation related to the project. Jira and Wiki was used to keep user stories and epics, as well as information about the project as a whole and information about the current sprint. In Confluence all the information across the teams were gathered like solution description, team routines, checklists, guidelines, retrospectives etc.
Jabber	Were introduced in a open-space session, which is similar to Twitter[2], however, without the length restriction. It was used for formal questions like questions regarding technical, but also for informal questions and activities like wine lotteries
Co-location	Both all teams and project management were located in the same location which were seen to be essential for good coordination

Table 5.3: Countiuned from Table 5.2 - Coordination arenas and methods

2:*Twitter is an online news and social networking service where users post and interact with messages, "tweets", restricted to 140 characters*

As seen in the tables, this very-large scale agile project applied many mechanisms for coordination which substantiates that coordination fast become complex as the size increases (Bick et al., 2016). Van de Ven suggests that one can divide coordination into three different modes, namely group, personal and impersonal mode of coordination, Section 3.2. The next sections will focus on these modes and what the findings from the case.

5.3 Impersonal mode of coordination

Impersonal mode of coordination is recognised by use of impersonally specified codified blueprints that determine an action like the use of plans, schedules, rules and policies in coordination. Examples of coordination mechanisms can be found in Section 3.3, Table 3.2.
As projects become larger and have to deal with more teams, the impersonal mode of coordination often become more important. Having some pre-defined structure on parts of the project is often more needed as that will ease the work progress later, and it will more likely get the project started. This way, one can minimise the likelihood of different interpretations of the same thing. However, the quotation below shows that this was difficult in practice.

> *'That is one of the places I think we were not good enough......We had not formalised the solution description, user interface, and behaviour good enough up front. Then when the different teams had made their part which was adequately related to what had been agreed upon; however, when you try to make them work together it did not work as it was supposed to'.* - Project Manager, Alfa

Although it is desired to have a project as agile as possible, some parts, especially when it comes to programming, and standards around implementation, there will always be different practices. That there are differences between teams should be taken into consideration to avoid extra work later.

Confluence[3] was used a lot for coordination, and was a platform where one posted information and documented all the things the team needed. One could find team routines, guidelines, routines across the teams, system documentation, checklists, etc. in the different spaces. To have a common area online across all the teams made the coordination easier, as team members knew where they could seek information. Also, the use of Jira[4] to keep track of tasks that needed to be done was seen as a valuable coordination mechanism.

> *'In Jira could we choose a delivery, and then click on it and see all the tasks that were to be delivered that period or iteration'* - Scrum Master, Gamma

The benefit of using a tool like Jira is that the information for the whole project can be gathered in one place. Jira was powerful to keep track of all tasks that needs to be done, and also being a powerful tool for bug- and issues-tracking. Together with a whiteboard for each team to keep progress status and schedule for what is being done 'right now' in a sprint was useful coordination mechanisms for the teams.

Although there were several guidelines regarding programming in the project there still were some challenges as an architect from Gamma describes.

> *'There were different views on how to use design patterns, some different habits on object oriented programming... and different opinions on how the guidelines should be followed'.*
> - Architect, Gamma

With teams from various contractors, the likelihood of different practices is high. There is a need to deal with this, and as the architect from Gamma also mentions, it is hard to know how much to plan up front as the project should also be agile. There are probably more challenges with this when the project is of very-large scale as there are more likely to be different habits and practices among the teams, and it also becomes harder to coordinate them.

[3] Confluence is a team collaboration software *https://www.atlassian.com/software/confluence Accessed: 15.05.17*

[4] Jira is an agile tool for bug-tracking, issue-tracking and project management functionality *https://www.atlassian.com/software/jira Accessed: 15.05.17*

5.3.1 Task uncertainty

Task uncertainty is one of the factors Van de Ven et al. (1976) mean has an influence on which coordination modes that are applied. Regarding the impersonal mode of coordination and task uncertainty, there were not that many relations, which it should probably not be either. The reason for that is that impersonal coordination mode is a more standardised way of doing tasks as well as plans and schedules. Thereby it should not be that many ways to interpret the same thing.
However, it seemed to be a challenge, especially across different teams, that people had differences of opinion on how things should be solved and how to follow the given guidelines. Thereby there were some uncertainties on how it should be done and different interpretations of guidelines/standards.

Even though projects use impersonal mode of coordination and have a plan, schedules and guidelines for how things should be done and to what time some factors influence the pace, for instance, task uncertainty. In the focus groups, there were mentioned some uncertainties around task difficulty, and the team's ability to manage tasks. Some teams used longer time than expected to complete tasks, while others were done much faster than anticipated. Uncertainties around the difficulty of tasks may have consequences like teams are unable to finish to what was expected, which leads to coordination issues when there are dependencies between tasks.

With a very-large scale agile project it is hard to have plans and pre-defined rules/policies for everything, as with impersonal mode of coordination. Thereby it is probably not the best coordination mode to deal with task uncertainty. It also weakens the agility of the project. However, it might be beneficial in the beginning as it may help to ease some challenges later in the project.

5.3.2 Task interdependence

There is an increased need for coordination when tasks are dependent upon each other. Task interdependence is the second influential factor Van de Ven et al. (1976) mention. Impersonal mode of coordination seeks at having clearly divided tasks with specific goals, and in Omega, all tasks were divided into user stories. Cooperation and coordination within the team were necessary to agree upon how to achieve all the user stories. Although there was an iteration board, where all the tasks had been broken down into specific tasks in a prioritised queue, you could not always follow it because of technical dependencies. This state that when a project becomes complex, there still might be other factors like dependencies which are difficult to anticipate that affect the progress despite plan or schedule that might exist.
In Omega, one could see the benefit of having impersonal mode of coordination to have a better overview of inter-dependencies among the tasks. The use of boards and tools like Jira resulted in the teams having a good overview of all the tasks. Then, it was easier to deal with eventual challenges along the process. Coordination was achieved by easier being able to reorganise what should be done next, both within the teams and across as an overview of all the tasks were accessible. Impersonal coordination mechanisms seemed to be increasing the effectiveness of the project.

5.3.3 Unit size

As a project grows in size and more teams are involved, it is more natural to have a greater need for more formal coordination, as it needs some more structure when dealing with more people. Unit size is the last influential factor which Van de Ven et al. (1976) point out to affect coordination modes, and impersonal mode of coordination seemed to be increasing in importance as size grew. Experiences from the Omega case, in general, was that both communication and coordination became more challenging when the number of teams increased, and it was more people to coordinate and relate to.

Although this was a very-large scale project with 13 teams at most, they consistently worked on optimising the project regarding coordination and structure between the teams as that created results.

> *'It is about optimising the overall nature of the project. The teams are better than the individuals, and the contractor team is better than the teams by them self. It is important to make the teams efficient'.* - Project Leader, Alfa

Compared to smaller agile projects, Omega experienced that a project of that large scale, requires more arrangements and guidelines, more leaders, and some more hierarchical approach to organising the teams. To achieve proper coordination in a very-large scale project it was experienced from Omega that to deal with unit size, you need more impersonal mode of coordination, as there are many teams to keep track of, and thereby some more standardisation is helpful.

Impersonal mode of coordination has through the Omega case turned out to be an essential factor to keep the project coordinated by having some administrative coordination as well as using technical tools to organize. Coordination in Omega seemed to benefit from using mechanisms from impersonal mode.

5.4 Personal mode of coordination

Personal mode of coordination is recognised by the use of mechanisms where individuals make mutual adjustments through vertical or horizontal communication, and examples of coordination mechanisms can be found in Section 3.4, Table 3.2.

The whiteboard was considered to be a significant artefact in Omega and worked as an information and coordination arena for the teams in relation to several coordination modes, and especially when it comes to horizontal personal mode of coordination.

> *'If you wanted to get an overview halfway through a sprint, you could just take a quick look at the left of the whiteboard to see whether it was much left to do'.* - Project Manager, Beta
>
> *'It was considered to be an important ceremony when you could move post-it notes. That "it is ready for testing", and something that has actually progressed. When someone is changing the status in Jira, it do not get the same attention, and thereby do not feel that important'.* - Project Manager, Beta
>
> *'A lot happened around the boards'* - Functional Architect, Beta
>
> *'It is the agile approach where the whiteboard is important, and to think loudly is essential'*

An important coordination mechanism, was for most of the teams to keep the plans and progress visible, and the boards reflected the present sprint queue. They were synchronised manually, but then again team members always knew what was going on, and it was easy to get an overview of current status of the project. Several agree that the boards have been a meeting point for discussions, informal conversations around the project, as well as a place for celebration as you have it visualised and physically right in the office. It also enabled visualisation of progress which eased coordination. The communication that happened around the board was valuable for coordination across the teams as well as within each team.

Several states the fact that once the whole project was co-located, the communication and coordination also became more fluent, and this was an important factor to success in this project. It becomes easier to seek out to the persons with the right competence, gain information, and then go back to your work right away which was an essential factor to coordinate efficiently with personal mode. One of the project managers at Alfa mentioned that sitting close to each other, as well as communicating a lot, led to better inter-team coordination.

> *'Everybody worked together across the teams/suppliers. We sat in the same location, very close to each other, and when the developers were using functionality that some other team had developed, they just dropped by them to ask questions and check up on things. It was a closeness, and people knew each other.'* - Project Manager, Alfa

As this project manager describes in the quotation, the simplicity of being co-located leads to more efficient usage of time. It gets easier to solve all those small problems, which often can be solved within two-five minutes if you are able to talk to those with the right competence and knowledge.

> *'It was easy to clarify orally, and it is easier to avoid the misunderstandings when you are writing things down and sending emails. You could clarify it right away.'* - Scrum Master, Gamma

Also, a scrum master in Gamma describes the advantages of being co-located and thereby being able to solve problems and ask questions to others quickly, makes the project more fluent, and it gets easier to make progress. It is time efficient to avoid writing emails back and forth since it is easy that you do not understand each other entirely from the first email. All this contributes to easing the coordination for the project as a whole as many issues get solved at a team level and with personal mode of coordination.
An architect from Beta describe meetings to be less important, and the regular face-to-face communication to be more relevant:

> *'I imagine that regular meetings are more important in the beginning, but become less important as one get to know each other, and gets more used to just seeking out to those you know can help you.'*

> *'That we knew each other, where people sat and what people did, resulted in easier being able to send people towards those with the right answers.'*

The informal communication is treasured when working agile and was an important factor in Omega. One can both receive and give relevant information just by communicating evenly, and it easier to pick up on things when informally interacting with others.

> *'The project manager dropped by the scrum masters at least once a day asking "How is it going, what is status, can I do anything to help?" Then, you could also get valuable information back as "they have solved that, they are doing that, or go and talk to him." Just having management by walking around, talking around was beneficial.'* - Project Manager, Alfa

By constantly giving small updates in person; one can coordinate without too much effort. This horizontal communication helped to decrease issues across the teams, as well as the project manager is able to keep track of who is doing what, and how the teams are progressing. Within the different contractors (Alfa, Beta and Gamma) there was some pair programming which enabled sharing of practices and standards within the teams which made the code more robust and similar. Also, it contributed to coordination across the teams, at least within the different contractors, which was one way of solving the issue with different practices, and enabled horizontal coordination.

> *'I think you need both (formal and informal communication), but the informal arena, if you had not had that dialogue - that people talk together without having both a common desire and capability to figure out and do things, I do not think large projects will work. I do not believe you can control and organise it good enough just by keeping everything formal.'* - Project Manager, Beta

> *'I experienced that the informal lines worked better than the attempts at creating places for people to meet, where things were discussed.'*

It was experienced, as stated in the quotes, that it is not the best and most agile way to force people to communicate and coordinate through meetings, but instead one should encourage the informal communication. Although the project is of very-large scale, the informal communication between teams helped the project to proceed faster, and many challenges can be solved at team level quickly. Personal mode of coordination was experienced as a critical enhancer to more efficient coordination. However, personal mode of coordination alone will probably not work for a large-scale project, but will surely ease some of the coordination challenges.

5.4.1 Task uncertainty

In relation to task uncertainty, there will, almost independent of what kind of project it is, be some uncertainties along the process. Personal mode of coordination has been seen valuable to deal with task uncertainty in Omega. The importance of mutual adjustments through communication both horizontally and vertically have been an important factor.
When a project becomes complex, it gets harder to make decisions and make progress. To succeed with personal mode of coordination, it is dependent on other people being available for clarifications when there are uncertainties around a task. In Omega it was also seen useful to know who has been participating in defining the solution description, so one could seek them to ask direct questions.

There were several discussions regarding uncertainties around the whiteboard, which has also been pointed out earlier to be a valuable artefact. One can achieve coordination by thinking aloud, thereby frictions and different perceptions could be clarified by communication between team members. That also opened for other relevant people, with right competence, to join in on the discussions which could lead to faster problem-solving.

> *'Once you start coordinating it is not that informal anymore, and then you miss some of the willingness the team members have to make their own day and make progress. You want team members to contribute to solve things themselves and deal with challenges at lowest possible level.'* - Project Manager, Beta

It was an ambition in the Omega project to try to let the teams deal with challenges at the lowest possible level, as can be read in the quotation. It is important that the teams feel empowered which often lead to higher motivation among the team members. By encouraging the teams to deal with issues themselves, coordination was achieved by mutual adjustments between teams and team members. With increasing informal communication, also across the teams, the coordination automatically increased as well. It became easier when trust was built, and people got to know each other better so you could just *'go over and ask which made it simpler to coordinate.'*

To be able to coordinate, when tasks are uncertain personal, (and group mode) of coordination have proven successful in this project. There was a lot of cooperation within the teams as well as across, and task uncertainty was dealt with through discussions and informal communication, which made the teams coordinated by themselves.

By having open discussions, it was possible to get several points of views, and maybe even input from others, which might have knowledge that the team did not have. Some mention that a possible problem with availability of people and "finding" those who have been contributing to detailing the solution description when it comes to personal mode of coordination. With increasing number of people in a project, it gets more challenging and complex.

The personal mode of coordination was maybe the most important regarding task uncertainty, especially when people got to know each other better, and it became easier just to go and ask people. This lead to fewer uncertainties and barriers and personal mode of coordination was found very efficient.

5.4.2 Task interdependence

Horizontal communication was experienced to be crucial for coordination regarding task inter-dependencies, as problems could be fixed and clarified much faster, and often right away. The informal communication was seen valuable to deal with task inter-dependencies since it was easier to get an overview of what people were actually doing when regularly talking to people. By knowing this, inter-dependencies became more visible and thereby easier to coordinate the other tasks around, and as a scrum master stated, it was easier to prioritise the progress further better. Since many tasks were overlapping and dependent on other parts, it was necessary that the developers went and talked to others to get answers. This seemed to work out well in the Omega project for coordination purposes.

Concerning task interdependence, all modes of coordination were visible through the analysed case and seemed to be necessary coordination mechanisms for success. However, personal mode of coordination, through horizontal communication, and mutual adjustments face-to-face, appears to play an important role - meaning the structure (organisationally) worked best when it was flat. A result of a flat organisation is that it is easier to make decisions and coordinate at a team level, and between teams, without having the hierarchical structure which can be time-consuming and not that agile. This is beneficial to deal with task uncertainty since it is often smaller problems or uncertainties which can be clarified fast at lower levels with personal mode of coordination.

5.4.3 Unit size

In the analysis of the coding, unit size in relation to personal mode of coordination gave most results of the different modes. Coordination seemed challenging when there are an increasing number of people in the project. How individuals are as a part of a team, as well as how team members interact with each other, have an influence on communication and thereby how successful the coordination is.

> *'I think it was many people to relate to.'*
>
> *'I cannot remember how many we were at the most, 100-150 people? And with shifting roles and people starting and leaving. You get exhausted. That is the drawback of large projects over a long time.'* - Scrum Master, Gamma

The first quote reflects what one of the involved in the project thought of the size of Omega. This is one person's opinion, but it is likely that more people felt this way during the project. The second quote, from a scrum master, touch upon the same challenge regarding the size of the project and to keep a relation to many people. It is very likely that there will be changes in the teams when a project is over four years, and if these replacements are too often, you often have to start from scratch to create a relation every time. This result in challenges regarding personal mode of coordination. That mode has been seen valuable for agile projects, but it is often built on a relation and trust which is built over time. Thereby, by often having to make new relationships can weaken that coordination mechanism.

When many people are involved, it is necessary to spread knowledge, and with the personal mode, one facilitates horizontal sharing which lead to enhanced coordination. One way of spreading knowledge and get coordinated between that many team members, which was used in Omega, were rotation of team members within the different organisations.

> *'You gained a lot of experiences by changing team. It was a pretty big difference between the teams and how they did things. Although the boards could look quite similar and every team delivered well, how they conducted things were quite varied. The people in the teams and how they were as persons, and as a team, had great influence. That was exciting to learn and experience.'* - Project Leader, Alfa

Coordination became easier as you got to know what others were doing, and also by knowing who were working on what and what kind of knowledge others were having. When the number of people in a project becomes as large as in Omega, it was also beneficial to get a feeling of how others are working and seeing things from other people's point of view. Otherwise it easy to end up with group-think[5] where you are "stuck" with your own solutions. It was thereby very advantageous to get new input and perspectives by rotation of people and from other team members stopping by. This rotation also helped to increase the personal mode of coordination as people got more confident in others and increased trust was built across the teams.

In Omega, the personal mode of coordination was of increasing importance throughout the project and helped to deal with many coordination challenges. As the project progress and team members interact, the team cognition will continuously increase, which will enable more personal mode of coordination. This often lead to higher efficiency as for instance several mechanisms through impersonal mode, and also group mode, are quite time-consuming and demand more effort. These are more likely to be substituted by personal mode of coordination, as the team members are more confident in each other, and that often end up with quicker problem solving and efficient sharing of information.

5.5 Group mode of coordination

Group mode of coordination is recognised by use of mechanisms in which mutual adjustments occur across groups/teams, mainly through different kinds of meetings. Examples of coordination mechanisms can be found in Section 3.5 Table 3.3.
A member of the project describes the change over time during the project regarding group mode of coordination.

> *'We adjusted meetings after what was needed in the project. Some meetings lasted the entire project period, while other meetings were either introduced if necessary or dismissed if redundant.'*

As time went by, the group mode of coordination decreased a bit in importance for coordination, and meetings were more scheduled when needed. Several mention that it was good to have the project a bit more structured and organised in the beginning, and that it was not that necessary to have all the meetings later in the project as it was easier to coordinate yourself when you know whom to ask.

[5] Groupthink is a phenomenon when a group of people get together and start to think collectively with one mind. The group is more concerned with maintaining unity than with objectively evaluating their situation, alternatives and options *http://examples.yourdictionary.com/examples-of-groupthink.html Accessed: 15.05.17*

> *'What I found most important was the combination of the semi-controlled meetings and those that just emanated.'*

> *'When meetings were dismissed, it was because we managed to cover the sharing and need of information between the teams. Either with shorter meetings or the fact that we got to know each other better, so we just went and talked to the relevant people.'*

In the beginning, meetings were scheduled to cover information needs that were not covered otherwise. However, a number of meetings changed over time, and as for the last two quotations, it seemed to be easier for team members to take the initiative to make progress and solve the problems at the team level as they got to know each other better. The necessary information sharing was covered without having to have strict and scheduled meetings all the time, at least not for all parts of the project.

Coordination that occurs through meetings is necessary in very-large scale projects as there are many teams to keep coordinated and up-to-date. Figure 5.3 in Section 5.1, show that meetings happened at several levels. This hierarchical division of meetings was mentioned as beneficial and time efficient since it could have ended up in being too many people to relate to, and too meetings to attend.

> *'As for me, I think the daily scrum meetings, followed by scrum of scrums per contractor and after that meta scrum. This division made the information flow both ways, both from the teams and up, but also downwards. I think those were crucial that hierarchical division.'* - Project Leader, Gamma

There were, however, some differences in the conduction and organisation of the regular meetings, dependent on the team. For instance, a project leader for Alfa describes the difference between the planning meeting in their teams and within the Beta teams.

> *'They had it 1-2 hours and hoped that it would work out. While we had done a detailed and thorough design, done by the team at the beginning of the sprint, where we often used the whole day, and even until lunch on the second day if the user stories were complex.'*

This state that although there are the same scheduled meetings for every team, there will always be differences in practices across the teams - especially with different organisations involved, which is also mentioned in Section 5.3. The teams should be able to structure themselves as they want, as agile methods highly emphasise self-organizing teams. Thereby the coordination at team level may differ which may give consequences to the project as a whole.

Some scheduled meetings were introduced and tried out for some time, but did not actually work like *open space*. It was, however, dismissed after two-three times and one of the project leaders from Alfa describes it as:

> *'The responsibility of everyone, often end up in being nobody's responsibility. That is what open space suffers from. There were some good discussions, but all in all, I do not think we gained that much knowledge from it. Maybe we got to know each other a bit better.'*

This kind of meeting can be valuable for knowledge sharing; however, it suffers if nobody takes initiative which can be difficult when you have to keep focus on a big project. Then, it might feel like a bargain and unnecessary use of time instead. In a very-large scale project, it might feel a bit too much as there are enough meetings and mandatory practices to attend and follow already. This kind of scheduled gathering did not bring any particular value to the project either.

A brought up example in the focus groups, which concerns coordination at group mode, is the technical demo that was for each release. The team members felt that it was very various what they gained as output from the demo.

> *'The technical demo was not that productive use of time for all the teams.'* - Scrum Master, Gamma

The problem was the varied effort the teams had put into the presentation, and they mention that they should have been stricter on what was expected to present, so it would not be too much or too little. It varied a lot whether it was useful or not for the teams. A suggested solution was to divide the demo, to have one detailed demo for the primary target, and an aggregated demo to cover the need for information sharing.

5.5.1 Task uncertainty

At group level, the solution description meetings, across the teams, were seen as a vital coordination mechanism regarding task uncertainty concerning group mode. It was both a technical and functional meeting where you went through what should be completed and done across the teams; several questions were brought up, as well as coordination and administration of tasks.

> *'They worked on a solution description, and together they gained an exquisite overview of what was delivered across the teams, and that lead to some extent of coordination across the teams.'* - Project Leader, Alfa

By defining a solution description some uncertainties could be removed, or at least you knew whom to ask questions if you were uncertain about a task. Those who identified the solution description were also able to coordinate the teams and have some control over what was being done.

There were some internal differences in this project, as some of the contractors spent, for instance, more time on planning meetings than others. Regarding task uncertainty, to devote more time was seen beneficially from Alfa's point of view, as they meant it created a sound basis for a successful sprint.

> *'Then the team is coordinated already, you know where to start and what to do.'*
> - Project Manager, Alfa

With an agile project of very-large scale, one is sentenced to meet some uncertainties regarding the tasks along the way. Everything is not supposed to be planned and pre-defined when you are working agile, thereby there will be changes along the way, and deciding the path as you go. How you handle this in very-large scale, determine how well the coordination of the project will be. Group mode help to plan across the different teams, and may help to solve inter-team uncertainties around tasks that can be clarified through group mode.

5.5.2 Task interdependence

Related to group mode of coordination, regarding task interdependence, several of the scheduled meetings that were regular in Omega were important. For each sprint, it was a planning day where there was an own dependency meeting. There, dependencies were discussed, which eased the progress of the project later, as they somewhat knew what parts were dependent on each other. For the whole project, a solution description was identified, which gave a good overview and coordination across all teams regarding inter-dependencies.

> *'Several of the meetings were introduced since "this is our central area this period, then others should not work on the same area".'*

By having these kinds of meetings, which were introduced after need, one could avoid several possible coordination and dependency conflicts and were more likely to be able to work more efficiently. These meetings worked as a coordination arena across the teams, and were important to avoid overlapping work activities and made it easier to agree which way to continue the project most efficiently.

> *'In the scrum of scrums meetings, it was possible to present planned changes in case it interfered with some of the other changes that were planned to do.'* - Scrum Master, Gamma

The group mode of coordination was especially beneficial across the teams, to have some scheduled meetings to be able to catch up inter-dependencies which might be difficult to catch through personal mode of coordination. Also if there were bigger issues, they were solved by group mode of coordination.

5.5.3 Unit size

In relation to unit size, group mode was experienced as an important mode to ease the coordination. However, one should balance between group and personal mode of coordination to achieve self-coordination at team level, to ease the work at higher levels. The power of horizontal communication and coordination across the teams have been experienced to be crucial, and the agility you then achieve is essential.

> *'It was some roles, however, the communication was unbelievably tight.'* - Functional Architect, Beta

To be able to work agile, horizontal communication is valuable in large projects, as also described in the quotation. And this also applies to group mode. Horizontal communication worked well in Omega, as team members were able to communicate well despite roles or hierarchy. This brought many advantages to easing the inter-team coordination also across teams.

Further, for group mode of coordination, related to status meetings, experience meetings, etc. it was vital to consider whom to involve in the meetings so that they were relevant as stated out in the quotation.

> *'If you have exchange of experiences with people who do not care, there is no point.'* - Project Leader, Alfa

Meetings are necessary for coordination of the teams; however, people do not want to spend all their time going from meeting to meeting, and not feel that it is necessary or relevant. So to closely consider "who to involve" was essential, and that is also an increasing challenge as the project grew in size.

Group mode of coordination was important throughout the whole Omega project. However, the mechanisms used for coordination varied and were adjusted after need which several mentions to be advantageous.

Chapter 6

Discussion

In this chapter, the three factors will be discussed shortly first in relation to the modes and why they have been relevant for the different modes. Then, in Section 6.1, 6.2 and 6.3, the three modes from the Van de Ven model, which was presented in the previous chapter, will be discussed in relation to inter-team coordination and evaluated against the theory. After that, in Section 6.4 there will be an evaluation of the Van de Ven model in relation to the hypotheses presented in Section 3.6. Also, an evaluation concerning the research questions in Section 1.2 will be presented in Section 6.5. Lastly, some limitations of this research will be presented in Section 6.6.

That inter-team coordination is an important factor for success in large-scale agile software development is substantiated by several studies (Larman and Vodde, 2010, Melo et al., 2013, Scheerer et al., 2014), and this exploratory study is thereby an important contribution to getting a deeper understanding of how to deal with inter-team coordination in large-scale agile projects.

The three factors in the Van de Ven model; task uncertainty, task interdependence, and unit size, are identified by multiple to be essential for coordination. It has been mentioned in Section 3.3 where Kraut et al. (1995) have identified large size, uncertainty, and interdependence to require an extra need for coordination, and that it requires particular techniques to ease the challenges regarding coordination.

- Task uncertainty: Through the evaluated case it has been evident that to deal with task uncertainty in a very-large scale agile project, it was necessary with all three modes of coordination identified by Van de Ven; however, they were important at different phases of the project.
- Task interdependence: In Chapter 3 it is mentioned different classifications of coordination; however, what is noticeable, is that several sources suggest that *'if there is no interdependence, then there is no need for coordination either'* (Bouckaert et al., 2006, Osifo, 2012). That substantiates Van de Ven's model where task interdependence can be seen as an influential factor to coordination, and can also be found in theory by Lindvall (2004) (Section 2.3. Regarding task interdependence, there has been experienced an increased need for coordination through the analysed Omega case. Task interdependence is very likely to occur in large projects, and all the three modes of coordination are necessary to deal with it; however, they are most important at different stages of the project.
- Unit size: As pointed out in Section 2.3 by Bick et al. (2016), there will be increased complexity with increasing number of teams. Regarding unit size, impersonal and group mode of coordination have both been important in a project of large scale to be able to handle

> coordination, keep track of the progress of the teams, and of course keep up-to-date on what people are doing. From what was experienced in the Omega project, it seems like it was a higher usage of these coordination mechanisms than maybe it would have been in an agile project, with for instance three teams.

This substantiates the importance of evaluating how those factors influence the coordination modes. The three modes of coordination are a good division for dealing with those three factors, and the next sections will look closer at how one can deal with the challenges of inter-team coordination based on what was experienced in Omega and in relation to other coordination theory.

6.1 Impersonal mode of coordination

With a very-large scale project, there are most likely people with different background and from various contractors. The likelihood of different perceptions on how to best solve an issue, or the best way to develop a system, is high. There will be different views on best practices and how to spend the work time most efficiently. To ease the uncertainties it was seen helpful to use impersonal mode of coordination to create some standardisation for the teams to follow, and also have some framing and schedule around the work to be able to coordinate that many. The difficulty of impersonal mode of coordination is how much such be pre-defined and decided before the start of an agile project. If there are too many routines, guidelines and plans to follow it is easy to lose the agility, and team members may not either feel that empowered. That agility can be decreased if there is a need for more standards and structure is also identified by Saeeda et al. (2015) in Section 2.3.
However, to help ease the start phase of the project, experiences from the case identify it to be advantageous to have some impersonal coordination mechanisms from the beginning. There might be some uncertainties along the way, especially regarding how to follow the pre-set standards, but in Omega, much of the uncertainty seemed to be solved by themselves across the teams. What can be learned from this case is that impersonal mode of coordination might be quite important in the start phase of a project, but decrease a bit in importance for success in the later phases of a project. However, defining boundaries of the project from the start, so all have the same vision from the beginning can be solved by impersonal mode of coordination, and can relate to external coordination which Osifo (2012) identify for an organisation.

The impersonal mode of coordination is necessary at an early phase and can be substantiated with Espinosa's classification of mechanistic coordination (Section 3.3). Mechanistic coordination is defined by Espinosa to be *'coordination by a program or by a plan with the use of artefacts, processes and routines to deal with dependencies with little communication'*. By having some kind of standardisation, the project will be steadier from the beginning, and people will know where to begin without being dependent on other tasks in the start phase. This can be seen beneficial as the other coordination modes demand more effort from the team members. That is more likely to be achieved later in the project when team members are more comfortable with their tasks as well as more comfortable being part of a team, and also have more relations across the teams. Thereby, it is beneficial to avoid tasks being too dependent in the beginning, which is often easier to achieve than later by impersonal coordination modes, as well as increase the scheduled meetings (Van de Ven et al., 1976).

A project of a certain size demands more structure, which also the Van de Ven model of coordination outlined in Section 3.6. As Osifo (2012) states (Section 3.1), coordination plays a major role in almost all phases of a project, and thereby need extra attention when the size of the project increase as well. There are always different ways of structuring a project with many teams, and still be able to keep much of the agility and let the teams as self-organized and empowered as possible. The way you choose to organise the teams regarding leaders, team size and hierarchical division can, however, have a huge influence on both motivation of team members, and flexibility of the project. Keeping the teams self-organised is, however, identified by Eckstein (2016) in Section 2.3 to be challenging. In the Omega project, there were 13 teams at most that needed to be coordinated and make sure everyone got the necessary information. With that many teams, it is likely that they have different practices and work habits. If all the teams have full empowerment over practices and their part of the project, it is very likely to run into more challenges when everything shall be implemented together. However, as seen in Omega the need for some more impersonal coordination, compared to smaller agile projects, was necessary. To have some standards and rules to start with, made the beginning phase of the project much easier, and people had something to relate to from the start regarding work habits. This can be substantiated by Mintzberg's (1980) standardisation mechanisms for coordination, which also are described to be important in the initial phase of a project to work as guidance for what is expected of skills, work processes and outputs.

Inter-team coordination is affected by the size, which is also Van de Ven point out in the four relations that are listed in Section 3.2. The formation of sub-groups seems to be increasing, and more mechanical methods are used to introduce information. The agile manifesto, which is presented in Chapter 2, state *'individuals and interactions over process and tools'* and *'responding to change over following the plan'*. If a project of very-large scale then needs a lot of structure, organising, and plans to be able to keep track of everyone, and keeping the teams coordinated, it oppose the foundation of agile practices. Finding a balance of structure and agility is crucial. More people demand more coordination, which again demands some more structure and organising. Is the result of this then less agility, limited self-empowerment, and less self-structure for the development teams? The question is, is it possible to still keep a project agile if it is even larger in size than Omega or will it demand "too much" impersonal mode of coordination by having too many routines, meetings, and organising to keep everyone on the same track? There are some of the suggested frameworks, SAFe or LeSS, in Section 2.5, which attempt to ease organising and how to handle agility in large-scale. However, those kinds of frameworks come with other restraints again which might not be suited for everyone. In Omega, team members seem to be more tolerant with the project not being full agile, by having more impersonal techniques for coordination and a bit more structural leadership which encourage more impersonal mode of coordination, and some more horizontal personal mode of coordination which will be discussed next.

6.2 Personal mode of coordination

According to Van de Ven model of coordination, Section 3.2 as size increase, there will also be an increase in personal mode of coordination (Van de Ven et al., 1976). Increased size can lead to several challenges to personal mode of coordination. To facilitate personal mode one of the advantages in Omega was that all 13 teams were co-located, which several brought up, during the focus groups, to be a major factor. Even though the location was not ideal for that many teams, several mention that it was beneficial as it was always short distance to everyone involved in the project to get help. It can be difficult for a large-scale project to find locations that are big enough; however, the sub-projects should at minimum be co-located. Being located at the same place and the same floor will increase the personal mode of coordination.

Informal communication and coordination through personal mode have been essential to deal with task uncertainty and is also easier when the teams are co-located. To be able to make mutual adjustments and collaborate across the teams by horizontal communication, ease the need

for coordination. Also, it makes it easier for the team members to know how the project is going and feel empowered as a team member. Coordination that is achieved by personal mode can be substantiated by Espinosa's definition of organic coordination, Section 3.4. Espinosa (2010) defines organic coordination to be *'coordination that are achieved through feedback or mutual adjustments, which will say mainly communication and interaction'*.
Similarities to Espinosa's classification of organic coordination can also be found in the Omega case regarding the importance of horizontal communication, and especially the informal communication. Since agile teams are empowered to make decisions at a team level, the personal mode of communication might be extra important to deal with task uncertainty, as it is often other team members that are able to clarify if you just ask. There are a lot of competence and information within that many teams. By getting to know your other team members, also across the teams, you can gain a lot of knowledge by just interacting with them. By trying to deal with uncertainties around tasks at the lowest possible level, the time is often more efficiently spent, without needing to have meetings all the time. By empowering the team members to deal with the problems, it is likely to get more motivated team members as well, as they feel they are contributing to the overall of the project. Personal mode of coordination help to decrease task uncertainty, and solve challenges faster.

There are likely to occur uncertainties across the teams regarding tasks and also work routines. In Section 3.1.1, task uncertainty was identified by Kraut et al. (1995) to affect coordination in large-scale projects. It was stated that there are more likely to be unpredictabilities regarding software, implementation, and prioritisation within and across teams regarding tasks, as there are likely to be people with different opinions with such a large project. Further, Kraut et al. state that if the coordination is poor to deal with task uncertainty, this alone can lead to integration failure. This substantiates what can be seen in the Omega project that it is quite essential to have good coordination to deal with task uncertainty, and it is beneficial that it is mostly solved with personal mode of coordination. Van de Ven et al. (1976) also point out a greater use of personal coordination mode to deal with task uncertainty in Section 3.6. The advantage of doing this at the lower levels in a large-scale project is that you may avoid many meetings that might are unnecessary or more easily solved by horizontal communication face-to-face. Also, the communication regarding prioritisation and implementation is crucial both at team level; as people should be on the same page and have the same visions, but also at an inter-team level; as these factors are important for further progress of the project and should either be discussed with personal communication or through meetings.

That informal communication will work is not given in a large project. In Section 3.1.1, it is presented studies by Curtis et al. (1988) which state that it is more likely to be communication bottlenecks and breakdowns in large projects. Even though that was not the case with the Omega project, one should be aware of the likelihood of that happening is increasing with an increasing number of team members. Also Kraut et al. (1995), Section 3.4, state that *'when dealing with a high degree of uncertainty, the informal, interpersonal communication is valuable for both the team members and for the project as a whole'*. It is also pointed out that the personal communication that occurs across functional boundaries helps to deal with uncertainty. By allowing teams to self-organize, as agile practices encourage, one enables and facilitate problem-solving at team level. Self-organizing team is an agile practice that one should not dismiss even though it can be challenging (Eckstein, 2016). If the teams are not encouraged to work as they want, one loose much of the agility, which may lead to unmotivated team members. As some mention in the Omega case, team members have to see the benefit of things they or doing otherwise they can fast become unmotivated. Thereby, coordination by personal mode of coordination is beneficial as it is a more direct way of coordinating with relevant people.

Agile development was not that common when this project was conducted, which meant many of the team members, as well as the leaders, had not done an agile project before, which could be a challenge when starting with a very-large scale agile project. The size affects the methodology, and although some had worked agile before, it was still different and other practices that needed to be included. Through the analysis of the case, working agile in relation to the size has not been mentioned that much. However, regarding the size, several have said that the informal communication, which is essential for coordination, has been a bit more challenging as there are so many to relate to. Thereby, size may affect the amount of informal communication that is possible to achieve, which may weaken horizontal communication which has been inter-team coordination.

The Omega project was at most 100-150 people divided into 13 teams. This project was one of the first in Norway to be agile of that size, and it turned into a successful one as well. When a project is of that size there will be several task inter-dependencies, and personal mode of coordination through horizontal communication has helped with faster problem solving, and an efficient way achieving coordination.

However, can it be "too many" team members in a project and still be able to keep it agile? Personal mode of coordination, which builds a lot on relations and horizontal communication, has pointed out to be one of the crucial factors for success, and a huge contributor to dealing with coordination of the teams in this project. If an agile project were to be even larger, will you be able to have the same personal coordination that was essential in Omega? From the focus groups, there were some that mentioned, already in the Omega project, that there were *'many people to relate to'*. How many people is it really possible to have a relation till at once? And how will this have an influence on coordination and agility of a project? Van de Ven et al. (1976) point out that when size increases membership participation often decrease which can be very crucial, and also more mechanical methods are applied (Section 3.2). It can thereby seem like size influence the ability to keep a project sufficiently agile, and that adjustments in methodology and work methods need to be done.

6.3 Group mode of coordination

Group mode of coordination was seen beneficial to keep track of the progress of the different teams. Regarding this, matrix structure was valuable in Omega (described in Section 5.1). By having some kind of hierarchical division within the project, one could avoid too many meetings and too much mandatory. This was experienced to be profitable in Omega as one wanted to spend the time as efficient as possible. Then, if you had to go to meetings that were not relevant to your case, it can fast feel like unnecessary use of time. That should try to be avoided as much as possible, and instead, try to divide the meetings to be for those who are relevant. To disseminate status updates about the progress was done hierarchically in Omega. First, it was to have scrum meetings team wise, then scrum of scrums with one representative from the team (within contractor team), and then meta scrum which included the team leaders and project leaders. This way knowledge and information were spread horizontally and vertically, both ways. Several mention this to work very well for this project. Van de Ven et al. (1976) also substantiate this, that to decrease uncertainties, a significantly greater use of the group mode of coordination is needed.

To be able to coordinate across teams, the adjustments made through group mode of coordination have been necessary. With an increasing number of teams it is likely to be several inter-dependencies between tasks, and group mode of coordination stated by Van de Ven et al. (1976) in Section 3.6 to be of increased importance to deal with it.

In Section 3.5, coordination was defined by Malone and Crowston to be the process of managing dependencies among activities, which is an early definition of coordination, but still a very relevant one. To achieve proper coordination one should also try to make the tasks as independent as possible, especially in large projects, since one should not have to wait for others to do your tasks. Mintzberg (Section 3.2), also favour dividing the work into distinct tasks. He further states that through mutual adjustments and standardisation one can achieve coordination among the tasks, which also apply across teams by group mode of coordination.

Interdependent tasks influence group mode as teams need to coordinate across the different teams, and it is beneficial to try and keep the dependencies across at a minimum to be able to work more efficiently. Omega dealt with inter-dependencies by trying to break down the user stories as much as possible, and try to have the teams work on different parts of the project (at least between the various contractor teams). These small steps, to make the tasks as independent as possible, helped to ease the coordination of tasks and can be seen in relation to well-known coordination theories, and also substantiated by Van de Ven's *hypothesis 2* in Section 3.6. With a project of very-large scale, it is implausible to avoid inter-dependencies at all. However, by dealing with task inter-dependencies in the best possible way from the beginning, and be able to adapt to upcoming changes, are probably the best way to achieve coordination. Through personal and group mode of coordination, this is obtainable. Group mode, as well as personal mode as mentioned, have both been increasing in importance throughout the project to deal with task interdependence, and are seen to be even more efficient in achieving proper coordination across the teams. They have been critical coordination modes throughout the whole project since they enable coordination through horizontal communication, and faster problem-solving.

6.4 Evaluation Van de Ven model

In Section 3.2, we can read that the Van de Ven model can be used to examine to what extent task uncertainty, task interdependence, and unit size can predict variations in the use of the three modes for coordinating work activities. Following, this will be put in relation to what in actually observed in the Omega case.

It is observed, through the case, that impersonal mode of coordination decreases a bit with task uncertainty. If there are uncertainties regarding a task, these often needs to be discussed further, and are not a standardised factor from the beginning. Thereby one could observe that personal and group mode of coordination were increasing in relevance when task uncertainty was high. Those were efficient coordination mechanisms and helped to share knowledge to deal with challenges. Hoegel et al. state that to deal with complexities and uncertainties, there is a need to have *'a dynamic and vivid exchange of knowledge to solve problems and adjust for emerging changes'* (Hoegl et al., 2004). The use of personal and group mode of coordination was then also experienced to be useful mechanisms for this, and all this then support Van de Ven's *hypothesis one*, Section 3.6.

When it comes to task interdependence, it was generally observed in Omega, that more coordination was needed to deal with it. All coordination modes increased in importance, and it was seen that a more flat organisation structure eased the coordination related to task interdependence. Regarding the personal mode of coordination, the horizontal coordination then increased; however, the vertical coordination decreased a bit. That is natural as the team members were adjusting across the teams, and not in a hierarchical way. To deal with interdependent tasks there were also need for some specific practices through impersonal mode and not just adjustments with personal and group mode of coordination. However, the last two were experienced to be most necessary for coordination purposes when tasks were interdependent. This somewhat support Van de Ven's *second hypothesis*, Section 3.6. However, it was not that visible differences in increased use of personal and group mode of coordination as the hypothesis state. They were experienced to be similarly increased.

Van de Ven states that unit size could also predict variations in the use of the three coordination modes. This factor was maybe least evident through the analysed case regarding the different modes, but still, there were some notable findings. As the project grew in size, it was necessary to have some more formal coordination mechanisms, thereby the impersonal mode of coordination increased in importance. It is not possible to avoid some extra formal coordination when there are that many teams and people to keep track of. Thereby some more standardisation, guidelines and plans are necessary although it shall be agile. It was also experienced that unit size has an influence on personal and group mode of coordination through the Omega case. As also mentioned earlier, many people involved means many relations to keep, and thereby personal and group mode of coordination will become more challenging to achieve. *Hypothesis three* are also somewhat supported with the analysis of Omega; however, not as visible as described in Section 3.6. From the analysed case one could see that impersonal coordination mechanisms played a major role and that there was a need for more standardisation that regular agile projects. There was also a noticeable decrease in group mode of coordination as the project grew in size as several meetings were being dismissed.

Van de Ven state that some fundamental factors explain why there are different mechanisms for coordination within an organisation. These factors, as mentioned above, has all proven to be influential on inter-team coordination in Omega as well. It was also experienced that concerning these factors, there was a need for different coordination mechanisms to deal with them.

However, what is not covered that well by the Van de Ven model, but was evident in Omega, is the fact that these needs for coordination mechanisms also changed along the project timeline. Mechanisms to deal with e.g. task uncertainty are most likely to be different in the second month compared to two and half years into the project. However, the fact that there are different mechanisms needed for coordination within an organisation is very accurate. Theory says the same that was experienced in Omega, that the same tasks might require different coordination modes after time (Espinosa et al., 2004). That would be an interesting aspect of studying closer.

6.5 Evaluation research questions

In Section 1.2 a description of the problem, which this research is aiming at looking at, is outlined. The primary research goal is ***'How can inter-team coordination be achieved in a very-large scale agile software development project?'*** With the two sub-goals; 1) What are the challenges, and the consequences of these? And 2) How can one deal with these challenges?

Concerning the sub-goals, several challenges have been found through the analysis with Van de Ven model, and also some solutions on how this was dealt with in the Omega case. Since this is just one project's experience, it is not given that it will work for every case. However, the insight is always advantageous and can be used to study inter-team coordination in large-scale projects further.

One of the biggest challenges is the fact that informal communication is extremely valuable to deal with all kinds of problems, clarifications, progress updates, as well as getting team spirit and motivation. Informal communication turned out to be one of the most important coordination mechanisms in Omega. However, informal communication is not necessarily a coordination factor that is easily achieved. Several aspects influence it which might be challenging. One influential factor is that informal communication is built on relations, and when a project gets large, there are many people keep a relation to. Since the projects often last over a long time, people also come and go, which mean you constantly need to create new relationships. Similar to experiences in Omega, Boos et al. (2011) state that coordination is dependent on people being able to communicate, and communication is thereby a vital factor for coordination.

Another influential factor in informal communication, as well as relation building, is trust. Without trust, people will not share information or see the benefit of sharing. That trust is necessary for coordination has been identified by several researchers in Section 3.4 to be crucial, which substantiate the findings in Omega.

The two mentioned factors, relations and trust, are very dependent on the individuals in the teams, as well as their ability to build relationships. There is not a straightforward answer to deal with this challenge; however, what was experienced in Omega was building the right teams, move around people if the person chemistry is not ideal. Also to open up for arenas for the teams to get to know each other, also across the teams, enables relation-building which often lead to increased trust. It has also been important to get insight on what the other teams were doing to gain trust in their expertise, as well as being able to keep track of their progress. This makes it much easier just to go over and ask questions since you know what they have done, and are confident in their knowledge. So, to have more group mode of coordination, in the beginning, to get to know the others, will hopefully lead to increased personal mode of coordination later in the project.

Another challenge which was evident through the analysed case is the difficulty of keeping the agility and self-empowerment of the teams, as well as being fully coordinated and having work practices somewhat the same. The challenge was to find a balance between agility and formal structure, which can be seen as a combination of traditional and agile software development (see Table 2.1 in Chapter 2). It was important to not have too much impersonal mode of coordination throughout the project as team members stated that it was easy to then feel less empowered, which again often lead to people not willing to make that much effort to progress. With the number of team members, which are involved in a very-large scale project, there is a need for some more structure and organising to keep everyone coordinated, and to make sure the project is progressing as it should with the same interpretation of the goal. However, it should not disable or weaken the agility of the project.

In Omega, it was solved by more impersonal coordination, in the beginning, to get it started, as well as quite a lot group mode of coordination which included several scheduled meetings. This was working very well in Omega to establish work routines, get to know each other and get the project started. As the project progressed, these coordination modes were slowly replaced with increased personal mode of coordination which also enables more agility. With such a big, agile project, I think it is necessary to have some more structure around the project than with a smaller agile project. However, it is important to preserve the agile practices and the benefits they bring. That the organisation is structured in a way that facilitates agility is important. In Omega, this was solved by using a matrix structure. This lead to team members being able to focus on their work, and not having to attend meetings all the time. Anyway, it was always someone in the team that got knowledge and information of the progress of the other teams, and thereby it was possible to keep coordinated.

Group mode of coordination, which concerns scheduled and unscheduled meetings to make mutual adjustments, is an important coordination factor in large-scale projects. Many of the meetings in Omega were considered crucial for coordination purposes, especially across the teams. The challenge with this in such a large project is how to conduct them in practice. Who should be included, how often should the meetings be, and what kind of meetings are necessary? These are all important to consider since meetings should not be a bargain for the teams, as they want to spend their time as efficient as possible, but still, be updated on the other teams. In Omega, this was somewhat solved by a hierarchical division of meetings, as well as several meetings were dismissed as the project progressed. Still, some of the interview objects stated that it was varied what they gained in knowledge and information from several meetings that still were present, like the demo. This can be a consequence of not being too strict on what was expected to be presented at the meeting, and it should maybe be dealt with another way. The granularity of what was presented in several meetings was various, and not everyone felt that it was relevant or necessary. Agile practices often involve several status update meetings in different ways, but the question again, who should be involved, and at what level should one have these various meetings. As personal mode of co-ordination increased in usage, it was also easier to dismiss several meetings which enabled more agility.

One last challenge regarding inter-team coordination, which was experienced in the Omega project, is that there are different coordination needs throughout a large-scale project. And one has to be willing and able to change some work routines as the project progresses since there are various needs regarding coordination at the different stages of such a large project. However, several challenges come with this. First, team members must be willing to change their work routines somewhat to optimise coordination. Secondly, people must be able to acknowledge that other coordination mechanisms might be more efficient in that phase of the project, and must dare to say their opinion regarding it. For instance, as mentioned earlier, several meetings were dismissed in Omega as they did not give any value to the progress. If nobody dares to say: "this does not work, and are not bringing any value to the project", the project will slow down. So, to be open and willing to change practices, and adjust after present need was essential.

If the listed challenges above is taken into consideration together with experiences from the Omega case, combined with theory, inter-team coordination may be more likely to be achieved in a very-large scale software development project. One should apply and include coordination across the teams without influencing and preventing the evolution of the project. Beneath, it is listed the four factors that stood the most out to have an impact on inter-team coordination.

- **Informal communication:** To spread knowledge, to self-coordinate, and obtain experiences from others across the teams.
- **Standardisation and structure (to some extent):** To be able to have some similar work methods, and make sure the project starts at the same point with the same interpretations of the goal.
- **Hierarchical division of meetings:** To ease the amount of meetings, and for the team members to spend the time more efficiently, and still be coordinated and updated of progress across the teams.
- **Clarifications at lower levels:** To deal with uncertainties around tasks both within the team and across, as well as being able to give and get help regarding a certain activity of the project, to clarify at the lowest possible level, through personal mode of coordination, has been valuable. Encourage to make the team's progress themselves, and encourage problem solving at team level. Important to know who is doing what, and what competence and knowledge people across the teams are having.

6.6 Limitations of research

The results of this analysis are a contribution to further research within the topic of inter-team coordination. However, there are some limitations to this research that influence the results that are found in the analysis.

The Van de Ven model was used to analyse the case. That was a well-suited model for this purpose and this kind of project; however, the limitations of using one model, is that the results will mainly focus on the factors that are identified with a certain model. It might not be able to catch everything that influences inter-team coordination in a project. The factors in the Van de Ven model inevitably influenced coordination in the Omega project, but the case is just analysed from that angle. It would be beneficial to look further into the case, and possibly examined it with a different framing, and maybe some other results would have been clearer. Compared to the other coordination models mentioned in Chapter 3, Van de Ven was however probably the most suited one.

The results presented in Chapter 5 and in the discussion in this chapter, are based on what is experienced in the Omega case. That mean, although mechanisms to deal with inter-team coordination that worked for that particular project, it is not given that will work on all projects. For instance, personal mode of coordination has been experienced to be essential for inter-team coordination here. Then, there are assumptions regarding individuals and the chemistry among the team members that are critical to personal mode of coordination to work. How the teams are put together, and how well it will work in practice is crucial. One cannot force people to trust each other and communicate informally, it must happen naturally, which is maybe not the case for every project. It would be interesting to study further how agile projects in large-scale unfold if the informal communication is less in focus.
Another interesting case to study further is to analyse an even bigger case, to see then how the important personal mode of coordination comes into place. Or are there any size restriction on how big an agile project should be to optimise coordination mechanisms. More people lead to more relations to establish, and it would be interesting to study if it can be too many involved in a project, so it weakens the advantages of personal mode of coordination instead of strengthening.

Another limitation of this research is the lack of access to interview objects, and the fact that the researcher was not attending the focus groups, just got access to the transcriptions. There then might be misunderstandings of the transcriptions or interpretations made, that might not be the case. This has been tried to be avoided as much as possible by communicating with one of the researchers who was present and transcribed during the focus groups.

The conduction of the interviews also took place several years ago, and many practices and insight might have changed or evolved since then, which should also be taken into consideration. As more and more organisations are applying agile methods, new routines and experiences are most likely gained since this project. However, the theoretical grounding and understanding is still lacking.

The validity of the analysis is influenced by only one researcher performing the analysis; however, that researcher is not bias by the interview objects, and are neutral in the analysis. The external validity, which will say to what extent it is possible to generalise the finding, and to what extent the findings are of interest to other outside of the case is quite high, at least the last factor. This analysis should be of interest to several other researchers to gain a deeper understanding of the topic inter-team coordination with a analysis of a real case.
As the findings are based on the researchers interpretation of the Van de Ven model, as well as the transcribed material which might have had an influence on the results the internal validity is quite

low as it is not certain that isolated findings from this case and their associated cause/effect will give the same output for others investigating the same.
However, the coding analysis followed a certain model and were done thoroughly in several rounds which increase the validity of the results.

Chapter 7

Conclusion

By conducting an exploratory case study on a real-life case, and analysing the collected data using the Van de Ven model, this research has contributed to gaining valuable insight and knowledge regarding the topic inter-team coordination, and how it is achievable in large-scale projects.

Experiences from this case study is that size of a project has an impact on how agile methods are applied, as size influences the complexity of the project that again affects coordination. Some of the most significant findings in Omega are listed to give a clear overview of the results.

- Task interdependence and task uncertainty influence what coordination mechanisms that should be used, as the Van de Ven model states, and all coordination modes increased in importance dealing with these two factors.
- As unit size increases, it was experienced a higher need for impersonal mode of coordination to ease the complexity of coordination.
- The arenas of coordination and the methods used for coordination changed over time. The different coordination modes of the Van de Ven model were needed at various periods of the project, to deal with inter-team coordination at different levels. More impersonal mode (by standards/plans), and group mode of coordination (by meetings) were crucial in the beginning. Then, the personal mode of coordination, especially informal communication, increased in importance throughout the project. That resulted in several meetings being dismissed as they were not necessary anymore.
- There need to be a balance between central control and self-management, to keep organised and still be agile. The informal communication has been experienced to be profitable in achieving good inter-team coordination, as many challenges were then solved at lower levels and much more efficient. However, it is not given that informal communication will work that efficiently for coordination, especially not from the beginning, as trust and relations must be established first. Also, one is dependent on being co-located to optimise horizontal coordination.

Inter-team coordination is complex and dependent on many factors. It is evident that inter-team coordination cannot be solved through a single coordination mode as there are different needs throughout a project. By having several modes of coordination in a large-scale project, and also adjust the coordination mechanisms that are used both on the size and what phase of the development project one are currently in, inter-team coordination can be achieved efficiently.

Further research should try and evaluate an agile software development project of even larger scale, to see whether there are any restrictions on how large an agile project can be and still be able to keep important coordination modes as personal mode, which has been crucial in Omega. How dependent can one be on horizontal coordination in large-scale projects?
Another factor that would be interesting to study closer, within the topic inter-team coordination, is whether there are any differences in what coordination modes are needed throughout the project for the same type of task.

Bibliography

Bick, S., A. Scheerer, and K. Spohrer (2016). Inter-team coordination in large agile software development settings: Five ways of practicing agile at scale. In *Proceedings of the Scientific Workshop Proceedings of XP2016*, pp. 4. ACM.

Bjørnson, F. O. and K. Vestues (2016). Knowledge sharing and process improvement in large-scale agile development. In *Proceedings of the Scientific Workshop Proceedings of XP2016*, pp. 7. ACM.

Boos, M., M. Kolbe, P. M. Kappeler, and T. Ellwart (2011). *Coordination in human and primate groups*. Springer Science & Business Media.

Bouckaert, G., K. Verhoest, and E. Beuselinck (2006). Reforms of central government coordination in oecd-countries: culture as counterforce for crossnational unifying processes?

Cohen, D., M. Lindvall, and P. Costa (2004). An introduction to agile methods. *Advances in computers 62*, 1–66.

Cohn, M. (2007). Advice on conducting the scrum of scrums meeting. *Mountain Goat Software Web site*.

Crowston, K., K. Chudoba, M. B. Watson-Manheim, and P. Rahmati (2016). Inter-team coordination in large-scale agile development: A test of organizational discontinuity theory. In *Proceedings of the Scientific Workshop Proceedings of XP2016*, pp. 2. ACM.

Crowston, K., J. Rubleske, and J. Howison (2004). Coordination theory.

Curtis, B., H. Krasner, and N. Iscoe (1988). A field study of the software design process for large systems. *Communications of the ACM 31*(11), 1268–1287.

Deng, X., T. Chen, and D. Pan (2007). Organizational coordination theory and its application in virtual enterprise. In *Research and Practical Issues of Enterprise Information Systems II*, pp. 311–316. Springer.

Dickinson, T. L. and R. M. McIntyre (1997). A conceptual framework for teamwork measurement. *Team performance assessment and measurement*, 19–43.

Dietrich, P., J. Kujala, and K. Artto (2013). Inter-team coordination patterns and outcomes in multi-team projects. *Project Management Journal 44*(6), 6–19.

Dikert, K., M. Paasivaara, and C. Lassenius (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*.

Dingsøyr, T., T. E. Fægri, and J. Itkonen (2014). What is large in large-scale? a taxonomy of scale for agile software development. In *International Conference on Product-Focused Software Process Improvement*, pp. 273–276. Springer.

Dingsøyr, T. and N. B. Moe (2013). Research challenges in agile software development. *ACM SIGSOFT Software Engineering Notes 38*(5), 38–39.

Dingsøyr, T. and N. B. Moe (2014). Towards principles of large-scale agile development. In *International Conference on Agile Software Development*, pp. 1–8. Springer.

Dingsøyr, T., N. B. Moe, T. E. Fægri, and E. A. Seim (2017). Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*, 1–31.

Dingsøyr, T., S. Nerur, V. Balijepally, and N. B. Moe (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software 85*(6), 1213–1221.

Dingsøyr, T., N. B. Moe, T. E. Fægri, and E. A. Seim (2017). Exploring software development at the very large-scale: A revelatory case study and research agenda for agile method adaption.

Dybå, T. and T. Dingsøyr (2008). Empirical studies of agile software development: A systematic review. *Information and software technology 50*(9), 833–859.

Eckstein, J. (2016). Sociocracy: An organization model for large-scale agile development. In *Proceedings of the Scientific Workshop Proceedings of XP2016*, pp. 6. ACM.

Espinosa, A., F. J. Lerch, R. E. Kraut, E. Salas, and S. M. Fiore (2004). Explicit vs. implicit coordination mechanisms and task dependencies: one size does not fit all. *Team cognition: understanding the factors that drive process and performance. American Psychological Association, Washington, DC*, 107–129.

Espinosa, J. A., F. Armour, and W. F. Boh (2010). Coordination in enterprise architecting: an interview study. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pp. 1–10. IEEE.

Gonçalves, E. and E. Lopes (2014). Implementing scrum as an it project management agile methodology in a large scale institution. In *European Conference on Research Methodology for Business and Management Studies*, pp. 461. Academic Conferences International Limited.

Greer, D. and Y. Hamon (2011). Agile software development. *Software: Practice and Experience 41*(9), 943–944.

Highsmith, J. A. (2002). *Agile software development ecosystems*, Volume 13. Addison-Wesley Professional.

Hoegl, M., K. Weinkauf, and H. G. Gemuenden (2004). Interteam coordination, project commitment, and teamwork in multiteam r&d projects: A longitudinal study. *Organization science 15*(1), 38–55.

Kitzinger, J. (2005). Focus group research: using group dynamics to explore perceptions, experiences and understandings.

Klein, H. K. and M. D. Myers (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*, 67–93.

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

Kude, T., S. Bick, C. Schmidt, and A. Heinzl (2014). Adaptation patterns in agile information systems development teams.

Laanti, M. (2014). Characteristics and principles of scaled agile. In *International Conference on Agile Software Development*, pp. 9–20. Springer.

Larman, C. (2008). *Scaling lean & agile development: thinking and organizational tools for large-scale Scrum.* Pearson Education India.

Larman, C. and B. Vodde (2010). *Practices for scaling lean & Agile development: large, multisite, and offshore product development with large-scale scrum.* Pearson Education.

Lee, G. and W. Xia (2010). Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *Mis Quarterly 34*(1), 87–114.

Lehtimäki, H. (1996). Coordination through social networks.

Lindvall, M., D. Muthig, A. Dagnino, C. Wallin, M. Stupperich, D. Kiefer, J. May, and T. Kahkonen (2004). Agile software development in large organizations. *Computer 37*(12), 26–34.

Malone, T. W. et al. (1988). *What is coordination theory?* Citeseer.

Malone, T. W. and K. Crowston (1994). The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR) 26*(1), 87–119.

Marks, M. A., L. A. DeChurch, J. E. Mathieu, F. J. Panzer, and A. Alonso (2005). Teamwork in multiteam systems. *Journal of Applied Psychology 90*(5), 964.

Mathieu, J., M. A. Marks, and S. J. Zaccaro (2001). Multi-team systems. *International handbook of work and organizational psychology 2*, 289–313.

Melo, C. D. O., D. S. Cruzes, F. Kon, and R. Conradi (2013). Interpretative case studies on agile team productivity and management. *Information and Software Technology 55*(2), 412–427.

Mintzberg, H. (1980). Structure in 5's: A synthesis of the research on organization design. *Management science 26*(3), 322–341.

Moe, N. B., H. H. Olsson, and T. Dingsøyr (2016). Trends in large-scale agile development: A summary of the 4th workshop at xp2016. In *Proceedings of the Scientific Workshop Proceedings of XP2016*, pp. 1. ACM.

Moore, E. and J. Spens (2008). Scaling agile: Finding your agile tribe. In *Agile, 2008. AGILE'08. Conference*, pp. 121–124. IEEE.

Nerur, S. and V. Balijepally (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM 50*(3), 79–83.

Nerur, S., R. Mahapatra, and G. Mangalaraj (2005). Challenges of migrating to agile methodologies. *Communications of the ACM 48*(5), 72–78.

Oates, B. J. (2006). *Researching Information Systems and Computing.* SAGE Publications Ltd.

Olsson, H. H. and J. Bosch (2015). Towards continuous validation of customer value. In *Scientific Workshop Proceedings of the XP2015*, pp. 3. ACM.

Osifo, C. (2012). Organization and coordination. *Proceedings of the University of Vaasa: Working papers*.

Østdahl, A. (2016). Agile project management in large-scale software development. *Literature Review - Preliminary Research Project TDT4501 NTNU*.

Paasivaara, M., C. Lassenius, and V. T. Heikkilä (2012). Inter-team coordination in large-scale globally distributed scrum: Do scrum-of-scrums really work? pp. 235–238. ACM.

Rico, D. F. (2010). Lean and agile project management: for large programs and projects. In *Lean Enterprise Software and Systems*, pp. 37–43. Springer.

Rising, L. and N. S. Janoff (2000). The scrum software development process for small teams. *IEEE software 17*(4), 26.

Runeson, P. and M. Höst (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering 14*(2), 131.

Saeeda, H., F. Arif, N. M. Minhas, and M. Humayun (2015). Agile scalability for large scale projects: Lessons learned. *JSW 10*(7), 893–903.

Scheerer, A., T. Hildenbrand, and T. Kude (2014). Coordination in large-scale agile software development: A multiteam systems perspective. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pp. 4780–4788. IEEE.

Scheerer, A. and T. Kude (2014). Exploring coordination in large-scale agile software development: A multiteam systems perspective.

Schwaber, K. (2004). *Agile project management with Scrum.* Microsoft press.

Serrador, P. and J. K. Pinto (2015). Does agile work?—a quantitative analysis of agile project success. *International Journal of Project Management 33*(5), 1040–1051.

Smith, L. R. and U. Schwegler. 11 the role of trust in international cooperation in crisis areas: of german and us-american ngo partnership strategies. *Organizational Trust*, 281.

Strode, D. E., S. L. Huff, B. Hope, and S. Link (2012). Coordination in co-located agile software development projects. *Journal of Systems and Software 85*(6), 1222–1238.

Thompson, J. D. (1967). *Organizations in action: Social science bases of administrative theory.* Transaction publishers.

Tore Dybå, T. D. and N. B. Moe. Agile project management.

Van de Ven, A. H., A. L. Delbecq, and R. Koenig Jr (1976). Determinants of coordination modes within organizations. *American sociological review*, 322–338.

West, D., T. Grant, M. Gerush, and D. D'silva (2010). Agile development: Mainstream adoption has changed agility. *Forrester Research 2*(1), 41.

Zmud, R. W. (1980). Management of large software development efforts. *MIS quarterly*, 45–55.