Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Benjamin Behm

Large-Scale Agile and Lean Transformation in a Globally Distributed Organization - Case Ericsson

Master's Thesis
Espoo, November 20, 2014

Supervisor: Professor Casper Lassenius
Advisors: Maria Paasivaara D.Sc. (Tech.)
Minna Hallikainen QBA

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Benjamin Behm

Title:
Large-Scale Agile and Lean Transformation in a Globally Distributed Organization - Case Ericsson

Date: November 20, 2014 **Pages:** x + 107

Major: Software Engineering **Code:** T-76

Supervisor: Professor Casper Lassenius

Advisors: Maria Paasivaara D.Sc. (Tech.)
Minna Hallikainen QBA

The interest in adopting Agile and Lean in large and geographically distributed organizations is increasing constantly. Organizations are seeking alternative ways of working to improve their ability to operate in a fast-paced market. However, several problems may emerge when Agile practices and Lean principles are adopted in a traditional organization.

Despite the increasing interest, there are not much research on how Agile and Lean are implemented in large-scale and distributed organizations. Hence, the main goal of this thesis is to follow how a large, globally distributed and continuously growing product development organization at Ericsson adopted Agile practices and Lean principles.

The research approach of this study was a qualitative single case study. Three sources of data were used to analyze the case organization.

The results of this thesis encompass a set of challenges and success factors emerged from the transformation within the case organization. Additionally, the results illustrate the phases the organization has undergone during the transformation. During the transformation, the organization has introduced several changes including building a common backlog, implementing Agile practices at the team level, and building an active Agile and Lean mindset. However, the organization has faced several challenges which have been mainly related to lack of a process definition, weakly defined roles and responsibilities, lack of training and coaching, and change resistance. The success factors include, among others, setting up a new leadership team and an organization culture that empowers individuals.

Keywords: Agile, Lean, transformation, large-scale, global, distributed, software, development, challenges, case study

Language: English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tietotekniikan koulutusohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä: Benjamin Behm

Työn nimi:
Laajamittainen muutos kohti ketteryyttä ja Leaniä kansainvälisesti hajautetussa organisaatiossa - tapaustutkimus Ericsson

Päiväys: 20. marraskuuta 2014 **Sivumäärä:** x + 107

Pääaine: Ohjelmistotuotanto **Koodi:** T-76

Valvoja: Professori Casper Lassenius

Ohjaajat: TkT Maria Paasivaara
Yo-merkon. Minna Hallikainen

Kiinnostus ketterien ja Lean-menetelmien käyttöönottoon suurissa ja kansainvälisesti hajautetuissa organisaatioissa on tasaisessa kasvussa. Yritykset ovat etsimässä vaihtoehtoisia tapoja työskennellä, jotta ne voisivat parantaa kykyään toimia nopeasti muuttuvilla markkinoilla. Monia ongelmia kuitenkin esiintyy kun ketteriä käytäntöjä ja Lean periaatteita otetaan käyttöön perinteisissä yrityksissä.

Kasvavasta kiinnostuksesta huolimatta, ketterien ja Lean-menetelmien käyttöönottoa suurenmittakaavan hajautetuissa organisaatioissa ei ole paljon tutkittu. Tästä syystä tämän työn tarkoituksena on seurata miten suuri, kansainvälisesti hajautettu ja kasvava tuotekehitysorganisaatio Ericssonilla ottaa käyttöön ketterät käytännöt ja Lean periaatteet.

Lähestymistapa tutkimukselle on kvalitatiivinen, yksittäinen tapaustutkimus. Kolmea erillistä tietolähdettä on käytetty esimerkkiorganisaation tutkimiseen.

Työn tulokset sisältävät joukon haasteita ja menestystekijöitä, jotka esiintyivät muutoksen aikana esimerkkiorganisaatiossa. Tulokset esittävät myös vaiheet, jotka esimerkkiorganisaatio on käynyt läpi muutoksen aikana. Organisaatio on tehnyt monia muutoksia prosessin aikana, muun muassa ottanut käyttöön organisaation laajuisen kehitysjonon ja ketterät käytännöt tiimitasolla, sekä rakentanut aktiivista ajattelutapaa tukemaan ketteryyttä ja Lean:ä. Kaikesta huolimatta organisaatio on kohdannut huomattavia haasteita, jotka liittyvät pääasiassa puutteellisesti määriteltyyn prosessiin, heikosti määriteltyihin rooleihin ja vastuualueisiin, vähäiseen harjoitteluun ja koulutukseen, sekä muutosvastarintaan. Uusi johtajatiimi sekä yksilötä kannustava organisaatiokulttuuri ovat osa muutoksen menestystekijöitä.

Asiasanat: Ketteryys, Lean, muutos, kansainvälinen, hajautettu, ohjelmistokehitys, ohjelmistokehitys, haasteet, tapaustutkimus

Kieli: Englanti

Acknowledgements

I would like to thank my supervisor Casper Lassenius and advisor Maria Paasivaara for providing me with such an interesting case for my Master's Thesis and for their guidance and patience throughout the process. I would like to express my very great appreciation to Minna Hallikainen for her guidance and for supporting us to get touch with people in Ericsson. I wish to express sincere appreciation to many individuals at Ericsson, who let me to interview them and provided me with valuable information about the organization, which built the base for this thesis.

Espoo, November 20, 2014

Benjamin Behm

Abbreviations and Acronyms

GSD	Global Software Development
ASD	Agile Software Development
CI	Continuous Integration
XaaS	Everything as a Service
CoP	Community of Practices
PO	Product Owner
SM	Scrum Master
DevOps	Development and Operations
SSR	Subsystem Responsible
SSA	Subsystem Architect
QQS	Quick Quick Study
AS	Application Specification
FI	Feature Investigation
FCS	Feature Concept Study
TR	Trouble Report
XP	Extreme Programming

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter provides a short introduction of the thesis. The chapter begins with a motivation of why the subject is important in Section 1.1. Then, a short introduction to Agile and Lean in a large-scale and global organization is presented in Section 1.2. After that, a research problem and research questions are presented in Section 1.3. Finally, a thesis outline is presented in Section 1.4.

1.1 Motivation

Engineers are one of the most expensive creative people in many companies and therefore, their productivity should be maximized [28]. On the contrary, traditional software development processes have often seen rigid for complex projects. These processes produce unnecessary overhead in the production that ties the resources of the developers to the work that does not actually increase the value of the software. It has been researched that in average 30–50 percent of all features in a software product are unnecessary or add

overhead [23]. Additionally, Microsoft has reported that 30 percent of features change after they have been specified and 20–25 percent of features will be removed from the original plan [19].

Adoption of Agile software development (ASD) methods and Lean principles should result in better, faster, and cheaper software development [42, 80]. Hence, software development organizations are increasingly adopting Agile practices and Lean principles. By using ASD methodologies, organizations try to eliminate time-consuming and prescribed cumbersome processes that will not add value to the software [34]. Lean is usually adopted to complement ASD practices with the aim to optimize and streamline the whole organization and to eliminate waste that does not add value to the customer [61]. Rodríguez et al. [67] report that an organization they researched was able to increase the productivity by more than 30% in some areas in addition to better customer satisfaction after introducing Agile and Lean.

There is an increased interest among practitioners in the adoption of ASD methods and Lean principles in large and globally distributed software development organizations [45, 63]. However, Lean in software development has not been widely researched nor is there much understanding on how usable Lean principles are in software development [24, 67]. Neither are there much research on how applicable Agile and Lean are in large and global software development (GSD) or how to successfully adopt them in such environments [24]. In addition, there is lack of information about how to combine Lean with ASD methods [67].

1.2 Background

Large software organizations have been increasingly distributed in more than one location, usually on several continents. There are many reasons support-

ing this phenomenon, such as concern for cost, need for highly skilled resources and appropriate mix of expertise. [32] In recent years, ASD methods have been widely adopted among large and globally distributed software development companies [7] despite the fact that these methods were originally designed to be used in small and collocated teams [33, 79]. Many organizations are driven by demands for developing software faster and thus they are trying to imitate the practices that high-performing teams have been using. The high-performing teams try to get rid of overweighted processes that hamper their innovation and creativity. [10]

The path toward Lean adoption in software development started with ASD [24]. ASD practices have been focused on team and project level [24] and ASD methods have been challenging to scale up to the whole organization [80]. Lean thinking has emerged in software development to complement the shortages of Agile methodologies [62] and it is assumed to be mandatory to scale up Agile [1]. Moreover, the demand for cost savings has increased the interest toward Lean principles among software development companies [24].

However, there is not a clear way to implement Lean thinking in a domain of software development [67]. Adoption of Agile practices can be done directly at the team level, but adoption of Lean requires changes in the whole organization [80]. Therefore, challenges will emerge when Agile methods and Lean principles are adopted together in a large and globally distributed organization.

This thesis provides insights of how a large, globally distributed and growth organization with a strong background of the traditional development process has been conducting a transformation toward Agile and Lean organization. Approximately half a year prior to this thesis, the organization got off to a good start with the transformation and after that many changes have been introduced. The main focus of this thesis is to describe the transformation approach selected in the case organization and to illustrate what has been difficult and what has been working well during the journey.

1.3 Research problem and questions

This thesis aims to study how a large, globally distributed and strongly growing R&D organization adopts Agile practices and Lean principles. The transformation approach will be described in addition to emerged challenges and well tried practices. Furthermore, the underlying reasons for the transformation will be explained because a large-scale organizational change is not a simple process as it affects the whole organization and its people. As such, this thesis answers to the following research problem:

How are Agile and Lean adopted in a large and globally distributed software development organization with a steep growth trajectory?

The research problem is studied from the case organization's point of view and is further divided into the following four research questions:

- **RQ1:** *Why is the Agile and Lean transformation started in the case organization?*
- **RQ2:** *What are the key phases in the Agile and Lean transformation?*
- **RQ3:** *What are the impediments encountered during the Agile and Lean transformation?*
- **RQ4:** *What have been the success factors during the Agile and Lean transformation?*

The research questions are answered based on a case study conducted in a single R&D development organization at Ericsson and are discussed in detail in Chapter 3.

1.4 Thesis outline

This thesis is divided into five main chapters. Chapter 1 introduced the background of the topic as well as the research problem and research questions. In Chapter 2, the existing literature related to the topic is presented. The research methodology is described in Chapter 3 including detailed discussion about the research questions. Chapter 4 presents the findings of the thesis. Finally, in Chapter 5, the thesis is concluded by comparing the results with the literature, and discussing some limitations of the study and presenting how the subject could be further researched in the future.

Chapter 2

Related Work

This chapter presents the existing literature of the subject. The chapter begins with a short introduction in Section 2.1. Then, Section 2.2 describes the basis of Agile and Lean in software development. Third, Section 2.3 illustrates the motivational factors that drive organizations to adopt Agile and Lean. Fourth, the transformation approaches are presented in Section 2.4. Fifth, Section 2.5 presents the challenges organizations have encountered during the transformation. Last, Section 2.6 presents the success factors that have helped organizations to manage better in the transition.

2.1 Introduction

Most software development involves a complex and unpredictable problem with high change rate [43]. In traditional software development methods, such as the waterfall model [68], work is progressed through the phases that respectively include requirements specification, design, implementation, testing, and maintenance. This is not the best approach when requirements

change constantly or the developed system is complex [43]. Hence, software development organizations are increasingly moving toward Agile and Lean development that should ease the software development process by enabling faster development and allowing to respond for changes more rapidly. However, the adoption of Agile and Lean may emerge challenges in a large and distributed organization.

2.2 Agile and Lean in software development

2.2.1 Agile

Agile methods in software development started to emerge in mid-1990s and several new Agile methodologies appeared during this period. Each of them was aimed to deliver software faster with high quality, less waste and overhead. [47] ASD methods, such as Scrum [70] and Extreme Programming (XP) [5], emphasize lightweight software development [79]. The initial philosophy of ASD has been encapsulated into four values in the Agile manifesto as follows:

> *Individuals and interactions over processes and tools*
>
> *Working software over comprehensive documentation*
>
> *Customer collaboration over contract negotiation*
>
> *Responding to change over following a plan*
>
> — Agile Manifesto, 2001 [6]

However, even though the Agile manifesto presents Agile development as a simple and clear concept [17], a single universal definition for Agile methods in software development does not exists [17, 41]. Additionally, the Agile

manifesto says that the way software has been developed in the past should be changed. For instance, high quality software may emerge without extensive planning and command and control management. [73]

According to de Cesare et al. [20], the three most important Agile principles for organizations are: 1) frequent collaboration between business people and developers throughout the project, 2) achieving customer satisfaction through the early and continuous delivery of valuable software, and 3) face-to-face conversation is the most efficient and effective form of communicating to and within a development team. However, the Agile manifesto does not provide instructions how to use the principles behind the Agile manifesto in action.

According to Jalali and Wohlin [38], the most common Agile practices applied in software organizations are continuous integration (CI), daily standup meetings, pair programming, retrospectives, scrum of scrum meetings, and test-driven development. Moreover, Agile practices rely on organizational communication which encompasses informal, day-to-day interaction among organizational members [4]. Additionally, ASD methods rely on people and their creativity instead of formal processes. This allows to deal better with unpredictability of the software development. [15]

Practitioners suggest that the use of Agile methods can be successful if focused on human and social factors. Additionally, people should have faith in their own abilities in addition to good interpersonal skills and trust. [22]

2.2.2 Lean

Lean management philosophy focuses on increasing value by eliminating waste [51]. Waste is any human activity that requires resources but does not produce value for the customer [82]. Womack and Jones [82] have described five core principles of Lean as follows:

1. *Specify value:* Lean thinking should start with defining the value of the product in order to deliver only what a customer is willing to pay for. Everything else is waste and should be avoided.
2. *Identify the value stream:* Identifying the entire value stream, from the order-taking to the delivery, is the second step in Lean thinking. The whole value stream should be optimized so that obviously wasteful steps have been eliminated.
3. *Flow:* The third step is to redefine the work of functions and departments in order to create a continuous flow of activities and to eliminate disruptive activities.
4. *Pull:* The production should occur only when needed. This eliminates unwanted inventories and ensures that customers get what they want.
5. *Perfection:* The organization should not stop getting better but continuously improve previously mentioned four principles. There are always ways to reduce costs or mistakes or delivering customer value more accurately.

Lean in software development does not focus on the process itself but rather emphasizes principles to guide ideas and insights about a discipline. However, the adoption of Lean thinking can be a challenge for some companies. It requires a change in culture and organizational habits. On the contrary, by understanding the essence of Lean and adopting it can lead to major improvements in performance. [62]

2.2.3 Scaling Agile and Lean

Originally, ASD methods were intended to be used in small and collocated teams of 3–8 people working on a single project or product [33, 79]. Lately, organizations, including traditional enterprises, have became attracted to ASD methods and started to scale them up to the enterprise level [7]. Addi-

tionally, the adoption of Lean principles is a growing trend among software development organization [24]. The phenomenon is also perceptible among practitioners as the use of Agile and Lean in large and distributed software projects has got more and more attention among them [44, 45, 47, 73].

Agile methods can indeed improve customer satisfaction, productivity, and job satisfaction, but the difficulty of adopting them into large and complex projects is still a limitation [22]. If the whole organization is transformed to work in an Agile way, it is not enough to focus on a team and project level [41]. However, ASD methods are often focused on team level practices [24, 80] and the original assumptions of the ASD methodologies may not be applicable in large-scale product development [41]. Adoption of Agile practices in a large-scale software development organization may be hampered by different complexity issues. For example, there might be need for a more complex management structure or need to standardize best practices to avoid miscommunication. [10] Thus, ASD methods have been challenging to scale up to the organization level [80]. On the contrary, Lean thinking allows to approach the problem in a more holistic way and enables to understand the underlying problems more effectively [80]. Therefore, Lean is usually implemented in the context of ASD [67, 81] and has been adopted to complement shortages of ASD methodologies in order to scale up Agile [62].

Poppendieck and Poppendieck [62] remind that it is important to look for the balance point of the Lean principles and Agile practices when adopting any of them. Some may work in one organization, but not in another. Additionally, Lean thinking should be well understood at the management level and many traditional practices need to be changed to successfully combine the Agile and Lean [80]. Agile practices may also require some tailoring when implemented in a large-scale [3].

2.2.4 Agile and Lean in global software development

Global software development (GSD) [11, 31] is a related area to the large-scale, even though geographical distribution does not always indicate of large-scale [3]. The features of GSD are distance, time zone, and national culture [11].

GSD is catalyzed by several factors, such as ability to deploy talented people regardless their geographical location, decreased development costs, improved proximity to the market and customers, and ability to reduct time-to-market [11]. Therefore, software development is increasingly distributed to multiple sites involving different cultures [31]. Despite the fact that ASD methods are well suited when customers and developers are collocated and they can have frequent interaction among each other [9], several software organizations have successfully adopted Agile methods in distributed software development [77].

However, the physical distribution of project members has several effects on different organizational levels. First, how to divide work between sites and how to handle change resistance are problems that have to be solved at the strategic level. Second, the cultural diversification, such as the need for structure, attitudes toward a hierarchy, and communication styles should be taken into account when working in several sites. Third, due to lack of informal and spontaneous converses, some issues can go unrecognized for extended periods. Last, the coordination of the work and process management related issues can also be critical. [31]

Distribution of team members hinders the adoption of Agile practices. One main reason for this is that many Agile practices rely on effective face-to-face communication which is difficult to achieve when the team members are not collocated. [34] However, an organization can alleviate these challenges in Agile teams using several practices. Team gathering, exchange visits, informal meetings with off-site team members, and gradual team distribution

help to reduce cultural differences within teams. [35] Additionally, by using supportive tools, such as phone, teleconferencing, videoconferencing, email and instant messaging, the organization can maintain the communication despite the geographical distance [3]. Furthermore, strict communication policy, reducing the number of Agile meetings, having a key person attending all distributed meetings are a subset of the practices, which ease working in distributed teams [35].

Agile development in GSD engenders challenges, as Agile development is people centric with informal processes whereas distributed development relies more on formal mechanics [65]. Thus, it is evidential that Agile practices need to be tailored in global development, especially if part of the development is outsourced to an external party [4].

Despite the fact that some have gained superior results when applying Agile methods in a distributed environment [76], the results may not often be as impressive. As the organization grows and becomes more complex, the amount of challenges will increase when adopting Agile and Lean.

2.3 Motivation for transformation

In a large and distributed organization, an Agile transformation may be a long and complex process [42]. It is about continuous learning and improvement requiring hard work, intense focus, and strict discipline [69]. Therefore, the initiation of a large, time-consuming transformation process should be well motivated within the organization. However, several studies indicate that correctly implemented Agile practices improve quality and add value over the traditional, plan-driven approaches [72].

Organizations are driven toward the adoption of Agile and Lean by several factors. According to the existing literature, most often the motivation

emerges from business and process related issues.

One source of motivation for the transformation is business related issues. Surviving in the competitive and fast-paced market while being both flexible and adaptive to change has been seen as an important driver for the change in several cases [7, 8, 10, 13]. For instance, Cloke [14] says that teams were not able to reach a required level of collaboration and velocity to succeed in the fast-moving market and thus the organization needed better ways to work. Furthermore, Brown [10] justifies the transition with an urge to substantially improve the efficiency in the development. In several cases, the transformation was initiated because of the demand for cost savings [10, 30, 52, 62]. However, not always are there major problems that would force the transformation. For example, Rodríguez et al. [67] say that the company was driven by the desire to remain a strong player in the highly competitive industry despite the fact that the company was already doing well.

The another source of motivation for the transformation is process related issues. A release process has often caused issues, which has triggered the desire to change the development process. Goodman and Elbaz [27] describe that they had major difficulties in estimating integration and testing phases accurately and due to this the releases often delayed and the quality of delivered features suffered. Therefore, they were forced to find out an alternative way of working [27]. Hanly et al. [30] had a similar problem. The organization was dependent on waterfall development methods and had major problems to deliver new functionalities on time [30].

The growth of the organization can also engender issues. Fry and Greene [25] say that the release cycle lengthened when the organization grew. A long release cycle can hinder the integration process at the end of a release [42]. Additionally, sometimes it was time to switch from the traditional waterfall based methods because they could not support the work anymore [25, 36] and existing processes needed optimization with more pragmatic methods [36]. Furthermore, increasing amount of development teams in multiple sites may

force the organization to seek an alternative approach to develop software in a way that would support a team oriented approach and would be clearly defined [66].

There are many other motivational factors that drive organizations to adopt Agile and Lean. These are, among others, to compensate for an increasing shortage of talented employees [24], to get feedback from the customers more frequently [74], to better fulfill the needs of end users [13], to improve a requirements elicitation process [49], and to improve the quality of the delivered features [10, 25, 30, 52]. Additionally, Conboy et al. [18] mention that suppliers, consultants, partners, and customers are increasingly coercing companies to use Agile methods in order to align interorganizational processes.

2.4 The transformation approach

There are not a single way to conduct an Agile and Lean transformation as the selected approach and progress depend on the organization and the context in where the transformation occurs. However, based on the existing literature, there are two main approaches to conduct the transformation: a big bang rollout and stepwise adoption.

In the big bang adoption, the whole organization is transformed at once within a short time frame. For instance, one global software development company implemented a full-scale Agile and Lean rollout from a traditional waterfall development process within three months, during of which the Lean principles were used as key tools to communicate the value of changing current behavior [25, 53]. The organization justifies the big bang approach by avoiding disagreements within the organization and presenting strong determined action. However, they already had an extensive automated test system in place and the generality of the R&D organization was collocated, which

enabled the big bang approach. [25]

In contrast to the big bang rollout, most organizations choose stepwise adoption where changes are introduced in smaller steps despite it requires more time in comparison to the big bang approach. The stepwise transition is often started with a pilot project [10, 13, 27, 29, 39, 49, 56, 57, 65, 78] in order to evaluate Agile methods in practice and to identify the practices that can be adopted in the organization. Usually pilot projects are small with low risks, but Goodman and Elbaz [27] started piloting Agile methods in a large and business critical project to really put methods under evaluation and to get a committed organization and executive support. Some pilot projects even evaluated Agile practices in a more complex development environment, including having multiple teams in globally distributed environment [10]. Small pilot projects may not be as visible as larger ones in the organization and thus lack of management support. This may result in the twisted perception of Agile. [27]

After the piloting phase has ended, the next steps vary between organizations. Some choose to create a rollout plan to support the rollout process [10, 78]. However, a successful transformation requires many changes in the organization and everything cannot be planned beforehand [36]. Misra et al. [54] present that transition from traditional software development processes into ASD processes requires changes in organization culture, management style, knowledge management strategies, and development processes. However, Tian [78] suggests keeping Lean principles in mind when proceeding with the transformation—avoid implementing what is not really needed.

To be Agile depends on a culture. Without a proper culture, the organization cannot be Agile. [48] For instance, ASD methods are the most incompatible with the hierarchical organization structure [37]. The organization should be able to loosen its policies and procedures and switch to development driven working where teams are self-organized and collaborating together [54].

A common culture across different sites is very important, even vital. Agile

practices depend on frequent collaboration and a single team can even be distributed to multiple sites. [21] Thus, the culture should be common for everyone so that it would support the collaborative way of working [55].

Lean is also highly dependent on a proper organizational culture. According to Rodríguez et al. [67], individuals and teams should be respected and trusted. The role of managers and product owners is not to push individuals but empower and encourage the teams to make their own decisions. However, the culture is much more. It is about continuous improvement and involving individuals in the transformation. It is about being open for new ideas and being willingness to learn. [67]

Software development is a collaborative process [12]. In an Agile organization, the knowledge is usually in a tacit form [54] as documentation is often put aside over working software. However, transferring and sharing the information between all stakeholders is a demanding task [12]. Thus, the organization should establish a way to manage the tacit knowledge.

The development process cannot be heavily process-centric but more short, iterative, test-driven, and people centric [54]. To keep Lean in mind, the organization could start, for instance, from the basis Scrum and add new procedures gradually after identifying what is still missing supporting the larger organization [78]. Nottonson and DeLong [56] said that they implemented a company backlog—a prioritized list of the projects to increase visibility among all stakeholders. Additionally, Hallikainen [29] says that team facilities may need to be rearranged to support Agile and Lean way of working. Furthermore, Bass [3] and Paasivaara et al. [57] argue that product owner teams are necessary to manage the scale and complexity of product owner activities in distributed software development organizations.

Larman and Vodde [44] recommend to use cross-functional teams instead of component teams in Agile development. In component teams, team members are specialized in a single function or component, which result in waste in the development process as a single team needs to hand off their work to the

next team. Having cross-functional teams decreases the lead time and waste as unnecessary handovers and queues between teams are removed. [44]

Training is an important part of the transformation process. External coaches can be hired to guide the early phases of the transformation and to educate the employees [46, 56]. Brown [10] says that intensive coaching and support are essential during the first months, as lack of these can result in severe implications. Some in-house training and workshops could be arranged for developers to provide an overview of Agile practices. These workshops could be a good place for rolling some practices and techniques out. [66] Additionally, the upper-level management and other stakeholders should be trained [10].

2.5 Challenges in the transformation

In this section, we present the major challenges that have hindered the Agile and Lean transformation in organizations according to the existing literature. The challenges are summarized in Table 2.1.

2.5.1 Management and organizational

Transition to Agile and Lean requires essential cultural and organizational changes [40]. However, the challenge is that organization culture nor the mindset of people cannot be easily changed [26]. For instance, some organizations found it very difficult to give up the command and control management style [7, 80]. Additionally, upper management can feel uncomfortable when talking about transformation. They are usually concerned with an ability to deliver features to customers. Managers are more comfortable when developers have promised certain functionalities for them on a specific date. [16] In organizations which have been able to maintain good perfor-

Table 2.1: Challenges

Dimension	Factor
Management and Organizational	Lack of management support
	Combining Agile and Lean
	A misaligned organization
	An organizational culture and a mindset of people are difficult to change
People	Change resistance
	Lack of competence
Process	Agile and Lean are not adopted organization widely
	Unclear roles
	Lack of customer involvement
	The surrounding organization is not Agile
	All talented employees are in the pilot project
Technical	Working with a backlog
	Decomposition of backlog items
	Unyielding facilities
	Lack of face-to-face communication

mance and delivery cycle, it can be harder to convince upper management to switch from traditional software development approaches to Agile practices. In that case, upper management needs to understand that Agile methods can decrease delivery time and save resources. Management will usually be interested in any process that could help organization to save money and increase efficiency. [16]

The transformation should be supported by the management. Without a management support, the transformation is practically impossible to execute. [7] For instance, management is needed to finance the transformation. Lack of financial support may became a threshold issue if individuals and teams cannot receive appropriate training and coaching in addition to tools

and facilities. However, not every company cannot afford to hire an appropriate amount of coaches to provide hands-on training [18]. Lack of coaches can cause problems during the transformation. People should get enough support and coaching relating to newly adopted methods and practices. [53] Otherwise there is a risk that people do not understand the principles and values well enough or the change is not lasting. Additionally, the coaches should not be too dictating regarding Agile methods, as people may become reluctant to adopt the practices [7]. Furthermore, the management may be unwilling to invest in training if they assume they know enough about the methods being adopted. Benefield [7] describes that the management was resistant to try out Scrum because they had not been well educated on Agile values and they thought they know enough about Agile practices.

Agile is team oriented, which limits its suitability in large organizations. A single team rarely has a required power to influence in an organization. Additionally, roles defined, for instance, in Scrum are not usually enough in large organizations, but on the contrary, roles such as a line manager might cause difficulties and having both new and old management roles can cause unnecessary overhead. Thus, the management needs to have a strong understanding about Lean thinking in order to implement a sustainable change within the organization. [80]

2.5.2 People

Some level of change resistance will occur in every organization which is being changed. Agile practices such as open work spaces, pair work, and 40 hours workweek may lead to change resistance among some developers and managers [50]. Organizations with a history of individual offices may be highly reluctant to switch to open work spaces [29, 67]. Puleio [64] says that friction emerged as people from different backgrounds were pushed in a small space to work together for the first time. The situation became even worse when different practices were experimented and people had to adapt

constantly to a new way of working. However, according to Hallikainen [29], the resistance of changes in physical work environment is natural and depends on individuals.

Changing roles and responsibilities may also lead to change resistance. Benefield [7] told that some managers felt left out when teams became more self-organized. Moreover, organizational resistance may become the main barrier to other organization to adopt new way of working [28].

If the Agile methods are mandated top-down, a lack of motivation among developers to use Agile practices may occur. The adoption may be onerous, and complex and thus, people might be reluctant to change. [18] Additionally, people may be feared that the direct and constant communication and collaboration, such as daily standup meetings and the use of storyboard, could reveal their deficiencies [18].

The resistance may also emerge from partners and suppliers who might see Agile and Lean as a threat to their business. It may be challenging to reach and educate all stakeholders and get them understand the value of the new way of working. [7]

Working in an Agile team requires competence in a broad set of skills. For instance, soft skills are important in Agile teams, and a lack of soft skills may hinder the collaborative nature of Agile teams [18, 21]. It can be difficult to find people with necessary Agile skills and training of all the skills can be highly expensive. [18] However, the management needs to pay attention to soft skills in addition to hard skills when recruiting new people. [21]

2.5.3 Process

If the transformation is conducted stepwise and only a part of the organization is adopting Agile methods and the rest is not, problems may emerge [9]. Similarly, Rodríguez et al. [67] say that the whole organization should adopt

the same Lean mode. Otherwise it is difficult to drive the change forward, especially if some part of the organization has not been aligned with the Lean thinking. In some cases, the traditional process may overlap or conflict with Agile practices. For example, documentation may be needed, which clashes with Agile principles. [49] Additionally, the life cycle differs in traditional and Agile methods. Agile methods focus on short iterations and immediately delivering functionality while the life cycle is much longer in traditional methods [9]. Moreover, achieving the benefits of end-to-end flow may be difficult without a seamless organization [67].

Occasionally, roles engender problems when switching from traditional methods to Agile methods. Especially the product owner and its job functions are difficult to define because the job titles in large organizations and product owner activities are not always standardized [3]. Additionally, Agile methods require that teams collaborate closely with customers and an on-site customer is desirable. However, this is not always possible as a customer might not be willing to be involved. [21]

There are several areas within the organization that are difficult to adjust to fit in an Agile organization, such as human resource incentives, matrix structure and time-tracking systems [7]. Furthermore, it may became a challenge to scale the practices up to the organization level, especially outside of the development organization [52].

A pilot project can also engender problems when introducing Agile and Lean in the organization. Gandomani et al. [39] note that pilot projects may become problematic because they are usually staffed with the most talented and most motivated individuals. This can create a wrong perception of their capability to be Agile [39].

2.5.4 Technical

In a larger organization, technical issues may hinder the transformation and even disrupt work of the organization. For instance, Maples [52] says that working with a backlog engendered problems. Many participated in the prioritization of backlog items and a consensus was difficult to reach. This often resulted in a chaos that interfered development teams. [52] Another challenge experienced by Puleio [64] was a lack of experience in decomposition of large tasks into smaller ones.

A large organization with a long history might have physical limitations in the surroundings, which prevent the whole organization to be collocated. In addition, open working facilities may cause extra noise that may disturb someone and that should be taken into account during the transformation. [29]

Face-to-face communication is seen as the most efficient and effective form of communication in Agile processes [79]. However, geographical dispersion challenge communication as face-to-face communication may be totally lost [79] and communication become less frequent and more constrained [11]. The socialization of teams becomes more difficult due to the physical distance. Communication lost its richness as supportive communication tools have to be used. A time zone may aggravate the communication problems if there are a few or any overlapping natural working hours between sites. Differences in national cultures may engender problems within a team. Common problems are lack of cohesion, miscommunication, and mistrust. Especially nonverbal communication is challenging as it is shaped by the culture and may cause misapprehensions. Lack of face-to-face communication may exacerbate problems as nonverbal cues may be lost when using communication tools. [11]

2.6 Success factors in the transformation

In this section, we present the major factors that have led organizations to succeed in their Agile and Lean transformation according to the existing literature. The success factors are summarized in Table 2.2.

Table 2.2: Success factors

Dimension	Factor
Management and Organizational	Have a strong management support and commitment
	Focus on organizational culture
	Have a strong buy-in from all stakeholders
	Build an active Agile and Lean mindset
People	Invest in coaching
	Provide training for all stakeholders
	Be willing to learn continuously
	Acknowledge everyone
Process	Start with a pilot project
	Build up processes based on the actual needs
Technical	Have test and build automation systems
	Invest in collaboration and communication tools

2.6.1 Management and organizational

According to many researchers, a strong management support and commitment are vital to succeed in the transformation [2, 10, 21, 25, 50, 78]. Livermore [50] says that there was a significant correlation between management support and involvement and the success of Agile adoption. For instance, without management support, the highly important financial sponsorship of

Agile teams might not be possible to have. Teams should be encouraged for example to travel to visit team members at different locations and to attend training courses and conferences. [21]

The organizational culture correlates with the success of the Agile transformation [2, 21, 52, 55]. Misra et al. [55] have researched that the organizational culture should support trust, rapid communication, dynamicity in requirements changes, and a short feedback loop with customers. Thus, a strong buy-in from all stakeholders is vital to support the transformation [55]. Moreover, it is up to the management to establish a culture that fosters the new culture within the whole organization [21].

According to Tian [78], the organization should build the active mindset of Agile. The Agile mindset helps to turn emerged problems into opportunities [78]. Additionally, the organization should have a synergy between Agile teams and senior management through leadership and collaboration. The management should change their mindset toward leadership which encourage collaborative working. [21]

The management style and the organization culture are linked together. Lean and Agile are both unanimous about the issue that traditional command and control management approach should be abandoned [18, 80] and the decision making should be collaborative instead of authoritative [54]. The management should also adapt to the practice that the teams should take more and more decisions [48].

A committed organization and the buy-in of the leadership are vital in Agile adoption [10, 25, 27]. The management should be visible during the transformation process and provide its support [10]. Additionally, the management has an important role in fostering the new organizational culture that drives the teams and the whole organization to adopt Agile ideology [21]. Building an Agile mindset should begin from the day one in order to instill the Agile way of working and continuous improvement in people [78]. Senapathi and Srinivasan [71] discovered that without management support and confidence

in Agile methods, it was hard to motivate team members and the risk for unsustainable Agile adoption was higher.

The management should also be able to empower itself to challenge dysfunctional organizational practices and policies to enable self-organized teams [80]. Additionally, Vilkki [80] argues that organizational practices can be changed most effectively when the management team works together on the transformation.

2.6.2 People

People are the key force in an Agile and Lean organization. Hence, it is important to provide appropriate training for the development teams, business stakeholders, and management [10]. The management may require specific training in order to get them understand new roles and responsibilities in addition to the values and principles of Agile and Lean [7]. Additionally, intensive coaching and support are needed [2, 10, 40], especially during the first months [2, 10]. Using external trainers and coaches to train personnel helps to provide a foundation in Agile principles and helps other teams to absorb support more easily [25]. Mencke [53] says that their transformation would have been smoother if the organization had hired external coaches when the transformation began. Coaching the management may also help to minimize change resistance from their direction [7]. Therefore, the organization should invest in coaches if possible. With a larger amount of coaches, more teams can be coached in a shorter time and the practices can be scaled up faster. Additionally, if there is lack of coaches, it is impossible to work deeply with the teams. Thus, it is preferable to provide intensely coaching for teams rather that offering shallow coaching across many teams. [7] However, senior team members can act as a coach and educate other team members [18].

People should be encouraged to learn continuously. It is beneficial to establish a culture of experimentation, where people are allowed to make some

mistakes. [53] Each tryout of a new practice is an opportunity to learn and to improve [78].

Everyone should be acknowledged during the transformation. It is important to listen people and ask their opinions regarding the transformation. For instance, team members may have a different view of things in comparison to managers. Furthermore, Benefield [7] says that there might be people who are not willing or able to adapt to the new way of working. Thus, when conducting a transformation, it should not be assumed that everyone would agree with all adopted principles and practices. Additionally, Smits and Rilliet [75] say that a sense of urgency throughout the whole organization helped to minimize change resistance.

2.6.3 Process

Several researchers say that a pilot program helps the transformation to succeed [10, 13]. Piloting is an easy way to determine what to do and where to focus on [10]. Additionally, Brown [10] learned from the pilot project that coaching and education is essential. For example, the shift from schedule-driven development caused complications and Scrum practices required guidance and practice. Additionally, traditional project management practices needed reinterpretation to reflect the Agile practices, which was a huge step for many who had used to work in a traditional organization. [10]

When implementing new practices within the organization, it is advisable to build up processes based on the needs and select only those practices that are actually needed [9, 78]. New practices should be added gradually in order to avoid unnecessary waste [78]. Additionally, some tailoring is always mandatory when adopting Agile and Lean in a large organization. This ensures that the methods and principles fit the needs of the organization because each organization is different and there is not a single method for everyone. [49, 52]

2.6.4 Technical

The organization should invest in a robust CI environment in order to streamline the delivery pipeline and to provide instant feedback for developers [27]. Builds, installation and regressions tests should be heavily automated in a large and globally distributed Agile organization because challenges tend to cluster between units that require close collaboration [8]. A CI system allows to run check-ins, builds, and tests more frequently, which is vital in order to maintain short development test cycles [25]. A CI system enables to find incompatible changes faster that decrease the amount of waste [27]. Additionally, having CI and deployment systems can reveal quality issues [18] and bottlenecks in other parts of the organization [8, 10]. Moreover, a CI environment and automated tests enable end-to-end development that is an essential part of an Agile and Lean organization [58].

High-quality collaboration and communication tools are also mandatory [58]. For instance, video conferencing tools are needed to enable active communication with distributed team members.

Chapter 3

Research Methodology

This chapter presents a design of the empirical part of this thesis. First, the research approach used is presented in Section 3.1. Second, the research questions are presented and discussed in Section 3.2. Third, the case selection is justified in Section 3.3. Fourth, Section 3.4 presents how the data was collected for this thesis. Fifth, the data analysis process is described in Section 3.5. Last, Section 3.6 discusses the trustworthiness of the thesis.

3.1 Research approach

The empirical part of this thesis is a qualitative single case study. According to Yin [83], a definition of the case study is:

> *A case study (...) tries to illuminate a decision or set of decisions: why they were taken, how they were implemented, and with what results.*
>
> — Yin, 2009 [83]

The case study approach enables to retain a holistic view of real life events especially when a new phenomenon is studied [83]. The singe case study can be justified in this thesis because it is not widely researched on how large organizations can successfully conduct an Agile and Lean transformation in a distributed environment. The single case study enables an in-depth study of the phenomena in the case organization with several different data collection approaches. One of the R&D units of Ericsson Company was chosen as a case organization for reasons explained in Section 3.3.

3.2 Research problem and questions

This thesis answers the following research problem:

How are Agile and Lean adopted in a large and globally distributed software development organization with a steep growth trajectory?

The research problem is further divided into four research questions which are discussed below:

RQ1: *Why is the Agile and Lean transformation started in the case organization?*

The underlying reasons for initiating the transformation in the case organization are investigated based on the perceptions of the interviewees.

RQ2: *What are the key phases in the Agile and Lean transformation?*

The main phases the organization has undergone during the transformation are analyzed based on the data gathered from the interviews and internal documents. As large changes in organizations usually require several years, it is assumed that the transformation is still in progress.

RQ3: *What are the impediments encountered during the Agile and Lean*

transformation?

It is known that introducing Agile practices and Lean principles in a distributed and multicultural organization will be challenging. Therefore, it is assumed that several challenges have emerged during the transformation in the case organization. The challenges will be researched using all data sources. The challenges found will be then categorized and described.

RQ4: *What have been the success factors during the Agile and Lean transformation?*

Despite the several assumed challenges, some positive results and experiments may also be emerged during the transformation. These will also be researched using all data sources. The success factors can include, among others, some phases of the transformation that have been successful, or used practices or selected approaches that have worked well in the case organization.

3.3 Case selection

The author was a member of Software Process Research Group at Aalto University and the thesis was written as part of the Need 4 Speed (N4S) research program in cooperation with Ericsson. In the fall of 2013, the possibility to research an early-stage Agile and Lean transformation in one of their globally distributed product development organization emerged. Therefore, a single and information-rich case [59] was purposefully selected to be researched. The case organization provided an access to research a complex environment that would enable in-dept understanding of underlying issues.

3.4 Data collection

Three sources of data were used to analyze the case organization: the qualitative interviews, observations, and organization's internal PowerPoint presentations. The data collection phase started on October 4, 2013 and ended on February 28, 2014.

3.4.1 Interviews

The aim of the qualitative interviews is to collect experiential knowledge from the representatives of the organization in order to answer research questions presented in Section 3.2.

The major part of the data was collected via in-dept semi-structured interviews. A total number of 32 interviews were conducted at four different sites in two countries as illustrated in Table 3.1 and in Appendix A. All interviewees were selected with the help of the case organization representatives. Interviewees were selected from different roles, backgrounds and organization levels in order to gain as complete view of the situation as possible.

All interviews were conducted as face to face in the organization's facilities. The interviews were semi-structured and open-ended with a predetermined structure. A number of questions and specific topics were decided beforehand. The main topic was the same for each interviewee but the questions were adjusted based on their position and background. The interview templates used are presented in Appendix C. However, the nature of the interviews was casual and the interviews did not strictly follow the interview template.

The interviews lasted from 24 minutes to 107 minutes. First interviews took longer as more background questions were asked in order to understand the

Table 3.1: Interviewees and their roles

Role	Site A	Site B	Site C	Site D	Site E	**Total**
Team member	3	1	1	1	-	**6**
Product owner	2	1	1	-	-	**4**
Architect	-	-	1	1	-	**2**
Coach	2	1	3	3	-	**9**
Subsystem responsible	1	-	-	3	-	**4**
Line manager	2	-	1	-	-	**3**
Other manager	-	-	6	1	-	**7**

Note: Since the same interviewee can hold different positions, the sum exceeds the total number of interviews.

history of the organization and the starting point for the transformation. Later, the interviews were shorter because only the questions regarding the transformation were focused.

Each of the interviews was recorded and the records were transcribed by an external party. In most of the interviews, there were two interviewers from which one served as a main interviewer and the other took notes and presented qualifying questions and supported the main interviewer. The latter one was also a senior researcher.

3.4.2 Observations

During the data collection phase, four observations were conducted. The information gathered during the observations was used to support and complement the interviews. Appendix B shows more information about the observations.

The first observation

The first observation session was arranged at site A on October 15, 2013.

The day started at 9.00 and ended up at 16.05. The author of this thesis was the only observer in this session.

During the first half of the session, a singe development team was followed and observed. This contained following Scrum events of the team: a sprint review, a retrospective, and a sprint planning. First two events were located in the conference room and the events were observed by sitting around the same table with the team members while taking notes. The sprint planning was arranged in the team area, next to their Scrum board.

During the other half of the session, a weekly demo and a product owners' meeting were observed. These were also arranged in the conference room.

The second observation

In the second observation session, the author and the first instructor of this thesis participated and observed in a value workshop arranged at site A. During the workshop, a set of organization's values were worked on. The workshop lasted 24 hours starting on November 25, 2013, at 12.00 and ending on the next day at 12.00. The observers spent approximately nine hours there, six hours on the first day and three hours on the following day.

The workshop was held by the management and Agile coaches and it was aimed for the people on site A and B.

The third observation

The third observation session was arranged at site C on December 17, 2013, along with the interviews. During the session, a weekly demo of the organization was observed. The session lasted 30 minutes. According to the organization's representative, an invitation to the demo was sent to the whole organization. Based on the observation, only three sites (sites A, C, and D) were using a video conferencing system. The exact number of participants could not be recorded, but 15 participants were identified from site A, 11 from site C, and two from site D. An Agile coach from the site A was facilitating the demo.

The author observed the demo by sitting on the corner of the same conference room with the others. Subjects of the discussions and behavior of people were recorded by hand.

The fourth observation

In the fourth observation session, the first instructor attended and observed the second value workshop at site D on January 20–21, 2014. The workshop was arranged like the first value workshop but it was aimed for the employees on sites C and D.

3.5 Data analysis

All interviews were recorded, which were then transcribed by a professional transcription company. After that, verbatim texts were analyzed using a qualitative data analysis and a qualitative data analysis software ATLAS.ti. After the records were transcribed, the author analyzed the transcriptions by reading them through and grouping subtracted quotes into themes according to research questions. Four main themes were reasons, phases, challenges, and success factors. These were further divided into smaller subgroups. Additionally, notes were written while reading the transcriptions, which were used to guide writing.

3.6 Data validation

Different types of triangulation were used as suggested by Patton [59] in order to improve the validity of study. First, most of the interviews were conducted by two interviewees, from which one was always a senior researcher.

These qualitative interviews represented the generality of the data collection. Furthermore, interviewed people were selected from different roles and backgrounds. Second, as it is not recommended to conduct a case study relying on a single data source [83], data collected from the observations and the organization's internal presentations were used to support the information gathered from qualitative interviews. The documents provided more information about team structures and working practices. Third, one of the organization's representatives validated and accepted the results presented in this thesis. Additionally, a feedback session was arranged at site A in early 2014. In the session, the results of this thesis were presented to approximately 30 company's personnel from sites A–D. Furthermore, part of the findings of this thesis have also been published in a publication in which the above-mentioned representative served as a co-author. Last, the author together with the first advisor personally visited four sites (A–D) and conducted the interviews face-to-face.

Chapter 4

Results

In this chapter, we first present a brief introduction of the case organization in Section 4.1. Next, we explain the reasons for the transformation in Section 4.2. Thirdly, the main phases are presented in Section 4.3. Fourthly, the challenges emerged during the transformation are described in Section 4.4. Lastly, the success factors and positive findings are described in Section 4.5.

4.1 The case organization

Ericsson is a large, multinational telecommunication company delivering a variety of mobile and fixed network equipment, software and services throughout the world. Ericsson has more than 110,000 employees and it operates approximately in 180 countries. Through the 1980s and 1990s, Ericsson developed its capabilities in plan-based projects and became very successful in delivering projects on time using a process similar to the waterfall model [58]. During the last decade, Ericsson has shifted from the traditional waterfall

process to Agile development as the competition on the market has become fast spaced [58].

Ericsson purchased a XaaS based platform in the early 2010s. The platform has been integrated from several enterprise level solutions used in telecommunication industry, which are subsequently referred to as components. Some of these components are developed by third-party and some by Ericsson. The development of these components requires highly specialized expertise due to their complexity.

The platform had already been used by one company that provided it as a service to its clients. The platform was originally designed to meet the requirements of that single company. After the acquisition, the customer base had grown and at the time of interviews, the platform was used by tens of customers, some of them were large and global companies. The amount of the customers was expected to grow rapidly in the near future.

Before the acquisition, approximately 30–35 people, including external contractors and consultants, developed the whole platform. There were separate groups of people implementing different components but they were basically all working at site D, see Figure 4.1. However, some development was outsourced to consultants in site E. At this time, the platform was rapidly developed while people worked closely together mainly across the same desk developing, testing, and verifying the system. The developers did not follow any particular development process but they had a freedom to choose how to implement given requirements in a given period of time. However, documentation was put aside partially due to strict deadlines, in favor of deliver on time.

Along the acquisition, approximately 10 domain experts from the previous development organization were hired permanently, who then continued to develop the platform mainly at site D. These people worked in cross-discipline, component teams developing and testing their own features.

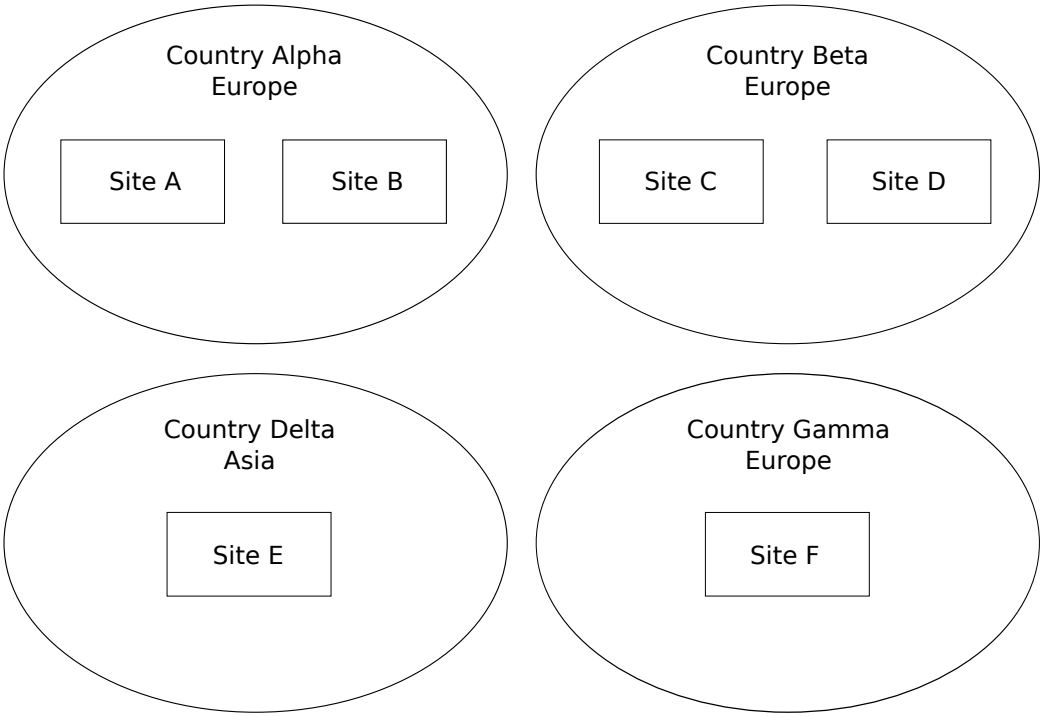

Figure 4.1: Sites

In late 2011, a small group of people started in country Alpha, who were distributed to both sites A and B. They formed two teams together with few experts in site D. Even then first steps toward Agile had been taken as these two teams adopted some Agile practices, such as a Scrum board and daily status meetings.

Since the acquisition, the first one and a half year was used to transfer knowledge from the old consultants to Ericsson's employees and to newly hired outsourced offshore consultants. The full-scale knowledge transfer began in the beginning of 2012, when a recruitement process was started in the organization. The whole 2012 was actually used to learn the system, get the knowledge from the existing consultants, to teach the new suppliers and to implement system improvements needed to stabilize it.

Since the acquisition, the organization had grown from a few dozen people to 200 people and from a few development teams to 15 teams. The growth

had centered mainly in site A partially because they had resources available with proper Agile mindset in addition to good facilities. Additionally, at the time of interviews, the organization was distributed to six main sites, as illustrated in Figure 4.1. The development was mainly occurred in five sites, A–E. The sixth site encompassed the customer support and the operations but it was not considered as part of the development organization.

4.2 Motivation for the transformation

This section presents the answer to the first research question that was:

RQ1: *Why is the Agile and Lean transformation started in the case organization?*

Six reasons for the transformation were identified from the interviews. The reasons are listed in Table 4.1.

Table 4.1: Motivational factors

Agile is part of Ericsson's strategy
To streamline the end-to-end flow
Component-based development was too rigid
The organization was not unified
The old process was not working well
To provide more visibility to the customers

Agile is part of Ericsson's strategy. One of the major drivers behind the transformation was that Ericsson aims to transform its software development organizations into Agile. Ericsson has successfully conducted several Agile

transformations in its other organization units, so it was natural to adapt the newly established case organization to an Agile way of working.

To streamline the end-to-end flow. One of the goals that the management had set for the transformation was to streamline the whole end-to-end flow as per the Lean principles. At the time of interviews, the flow went through several phases which were requirements elicitation from the customers, analysis and prioritization of the new requirements, implementation and testing of the features, and finally the deployment and delivery. This whole process had been highly waterfall driven with several steps, freezes, and handovers as illustrated in Figure 4.2. The aim was to remove non value adding procedures from the flow and to decrease the total lead time.

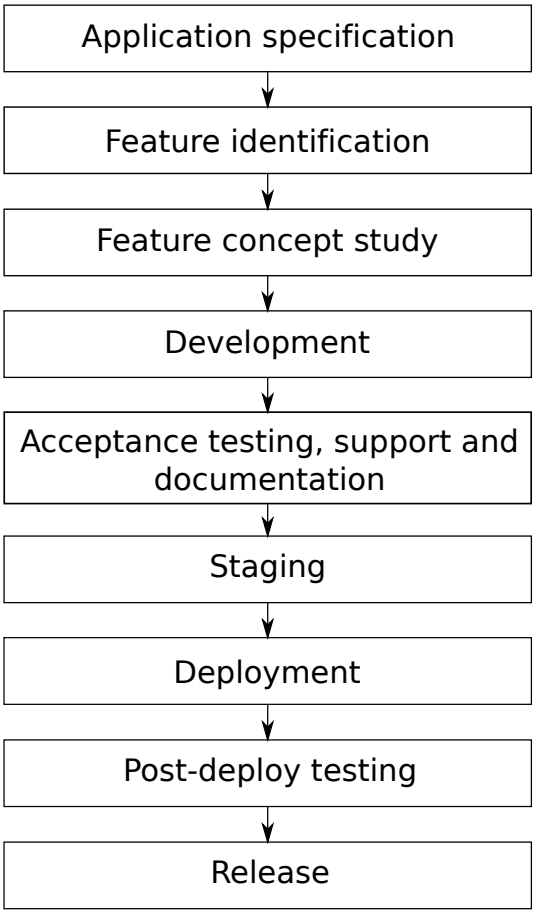

Figure 4.2: End-to-end flow

The first phases of the flow had usually been conducted by the requirement team and the portfolio manager together with the product owners. The flow has been simplified as follows: 1) First the requirement team comprising system architects and managers decided what requirements are taken under deeper evaluation based on customer negotiations. 2) Then an architect and a system team, called the system core, studied the dependencies of each requirement within the whole system. The information was then passed to the product line who then decided whether the requirement will be implemented or not. All accepted requirements were then put in the backlog. 3) Then the portfolio manager together with the product owners estimated needed resources and hours for each component and made a plan how to implement and deliver the requirement. After that, teams were able to pull features from the backlog. However, the features were often so large that they could not be implemented within a release. In addition, the estimates of needed man hours were not accurate at all.

At the time of interviews, the release cycle was eight weeks. Only first three or four weeks were used for coding new features in teams' code branches. After that, the code base was frozen and new features were not usually allowed, only found issues related to the newly implemented features could be fixed. The last weeks were spent on testing, verifying, and integrating the code into the releasable system. The development teams were needed to fix found issues and to write appropriate documentation to support testing and deployment. As seen, plenty of time was needed for testing and integration because appropriate automation, regression testing and continuous integration systems were missing. This was a major bottleneck as it limits the efficiency of the teams to implement new features.

Additionally, the business was always stopped for a few hours at the end of each release cycle. This meant that the service was not available during that time. Organization's goal was to be able to do hot deployment, meaning that developers could deploy new features and improvements into the production

without shutting down the servers.

Component-based development was too rigid. The component-based team structure was established in early 2012. Each of the existing people was a component specialist and new people were hired with specific component knowledge. At this time, a single team usually comprised 10–20 people and some of the teams were distributed to multiple sites.

The component teams were not efficient at the development and lead times used to be long. A single feature was implemented sequentially a component at a time by separate component teams. Each of these teams might have located in different sites, for example, a front-end team at sites A and B, a testing team at site D, and database team at site E. Therefore, it was challenging to try to plan and coordinate the whole development process. Teams had to align their tasks with other teams so that the developed features would have been finished on time. Sometimes unfinished features mired in a queue for a long time as the next component team was busy with another features.

As the organization grew and features started to increasingly depend on several component areas, the coordination of work became even more challenging. Furthermore, teams had difficulties in finishing promised features and the management was under a high pressure due to that.

The organization was not unified. One reason behind the transformation was lack of inter-team spirit in the organization. Teams tended to do a site-specific sub-optimization rather than working together over sites. Furthermore, each site had a certain functional area to focus on. Sites A, B, and E focused on development, while site C had the system knowledge, site D focused on testing and site F on deployment. Additionally, the growth centered around site A. As a consequence, some friction had engendered between sites and the management wanted to eliminate it. Therefore, the goal was to unify the organization and to narrow the gab between sites and to bring everyone including product line, R&D, and operations closer to each other by increasing collaboration and communication between sites.

The old process was not working well. The previously used process was not well defined nor managed and hence there was an urgent need to improve that. Some developers even mentioned that they had no process at all, despite the fact that Ericsson had successfully implemented a sequential process that it had been using decades.

> *As I saw it, we had no process in the beginning that we would've been following... So no Agile processes, nor traditional waterfall models.*
>
> — Team member

Many decisions were dictated by managers but sometimes they had different opinions on what features should have been implemented and in which order. Occasionally, teams were asked to switch to another feature by another manager even though the previous feature was still in progress.

> *Decision was taken.. on management level, for everything. And, some manager even, did go directly to the team and said: "You should do this, please add this to the release". But we don't have time. "Add it". So it was quite much a management that there also, interfered.* — Manager

To provide more visibility to the customers. Due to several backlogs and inaccurate requirements management, accurate information about the state of requirements could not be provided for the customers. However, a roadmap with requirements to the next few releases was provided for the customers, but there had been problems of getting promised features out on time.

In addition, customer representatives communicated only with a few key people in the product line, who had not always got the latest information about the progress of the requirements. There had not been a clear communication channel between the customers and development teams.

4.3 The transformation approach

This section presents the answer to the second research question that was:

RQ2: *What are the key phases in the Agile and Lean transformation?*

Figure 4.3 illustrates the main phases that the organization had undergone since the platform was acquired. The most important phases are described in the next subsections.

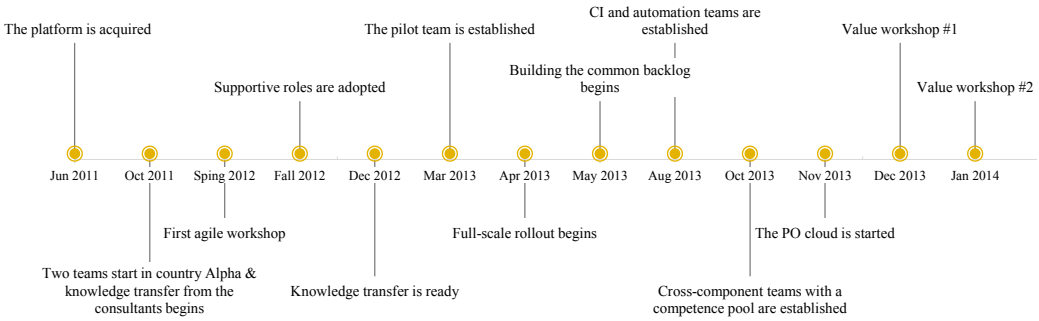

Figure 4.3: Timeline

4.3.1 Component-based organization

The organization had started speaking about Agile and Lean already in the late 2011. For example, there were a few teams that had adopted some Agile practices. Additionally, the first Agile workshop was arranged in the spring 2012. However, the organization decided to start a component-based team structure in the early 2012. The main reason for this was that each component required highly specialized knowledge and it was time-consuming to learn even a single component.

> *You cannot really ask people to learn more than one component in one two years.* — Product Owner

In those days, the organization started to increase the amount of people mainly at site A by hiring new experts externally and transferring people internally within other organization units. These people were divided into component teams based on their competence. Most of these teams had been distributed to several sites with approximately 10 people in each team.

By the middle of 2012, the system had expanded to a great extent and features had started to depend on several components. It was noticed that someone should watch over different components and prioritize tasks. Thus, the product ownership was taken into use and a team of area product owners was established. Each product owner got a single component to take care of.

In parallel with the product ownership, virtual feature teams were introduced. The virtual feature team was set up for each feature affecting on multiple components. In practice, a required number of resources was picked from the component teams so that the newly established virtual team would be able to implement the feature, as illustrated in Figure 4.4. The team was usually dismantled after the feature had been finished unless the same team would have been able to implement the next feature with the same constitution.

However, the development in and the coordination of the virtual feature teams were not easy. Setting up a virtual team was a complicated task as not always were people with right competence available for a specific feature.

> *Setting up the virtual teams was challenging because we had the feature and then we found three guys [with competence A] but we don't have [competence B] because they're all busy with other features. So here we have the resources [with competence A] available but then we cannot wait for three weeks, so the guys start with something else. So it's like a puzzle all the time.*
>
> — Manager

Additionally, the team members thought that it was challenging to work in virtual teams. The surrounding people changed constantly and it took time

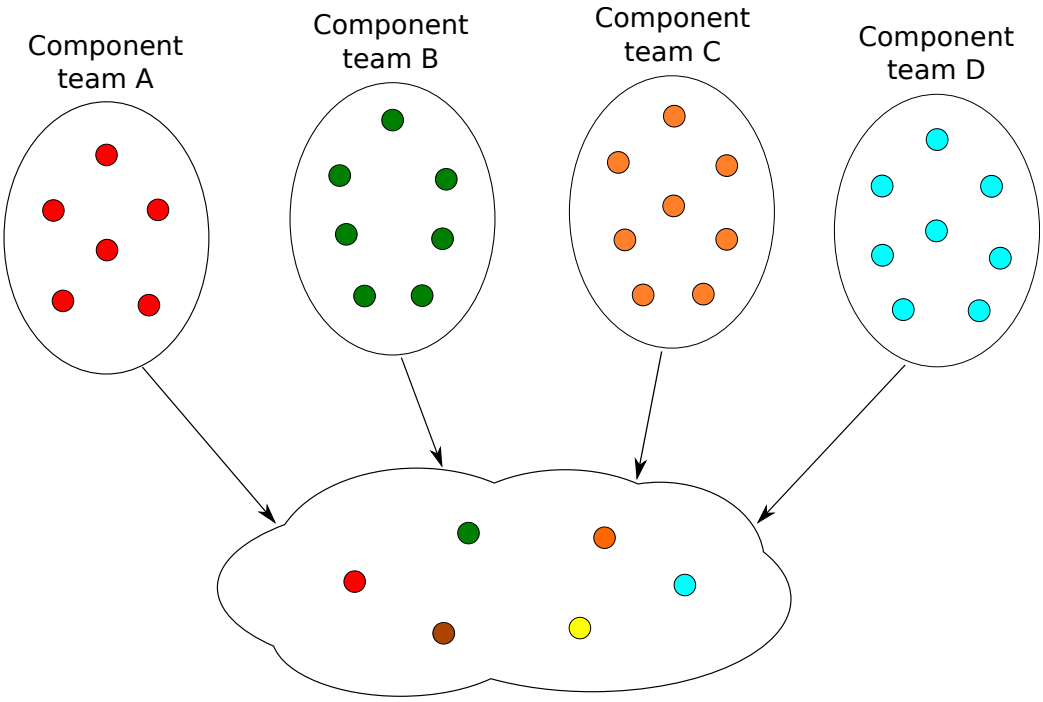

Figure 4.4: Virtual feature team

to make the acquaintance of new people, which hindered the trust and team building.

4.3.2 The change vision

In early 2013, the leadership team together with the Agile coaches arranged a three-day workshop, during of which they produced a prioritized list of subjects that the organization should focus on during the transformation. They came up with the concrete ideas of what to do next such as to establish a common backlog for the whole organization and to start the value discussion in order to improve the affinity between employees in all sites.

In addition to the foregoing subjects, they produced a vision for the organization. The vision was called “The Showcase” and it illustrated the desired state of the organization in two years and described the action steps that

were needed in order to change the organization.

However, a strict plan was not made. The idea was to introduce changes in small steps without doing extensive planning beforehand.

4.3.3 The common backlog

The common backlog was one of the topmost improvements in the list of the desired changes and therefore it was started first. It would support the new Lean ideology and would help the organization to streamline the end-to-end flow. It was also desired to improve visibility and help to define the lead time of new requirements.

Earlier, several different backlogs had been in use. An electronic backlog management tool was only used for issue tracking and different stakeholders had their own Excel Spreadsheets to manage requirements, features and improvements. This hindered the transparency and made it impossible to define the cycle time of a single requirement and to see the whole picture of the end-to-end flow.

> *We had Excel documents that had a big plan, all the features and all the improvements to be developed in Excel.* — Manager

Building the common backlog started in early 2013 and it was finished in the summer of 2013. Nowadays, every new features and improvements will be added to the single backlog. However, the common backlog was only for high-level features and improvements. Each team had a team backlog in where chosen features and improvements were split into smaller user stories.

4.3.4 The pilot team

The organization established a pilot team in early March, 2013. The aim was to evaluate how a totally cross-component and cross-functional Agile team

could work. Forming the pilot team started by finding volunteers who would like to take part in the experiment. Five developers were found in site A and two architects in site D. Each of them was highly experienced and had specialized in different components and had a positive attitude toward an Agile way of working.

However, the existence of the pilot team engendered problems in the whole organization and disrupted the work of other teams. The volunteers had had a central role in their component teams and their absence interfered with the work of those teams. Teams lost key people with important knowledge and a few other people became overloaded because of increased inquiries. At that time, teams were not efficient in finishing features on time and the management was under considerable pressure.

The pilot phase lasted only a few weeks due to issues occurred. Even though the team would have been really competent, the management decided to dismantle it and the transition to cross-component and cross-functional teams were started in full-scale. In consequence, the team members from site D were replaced with two new team members from site A and that team became the first new cross-functional and cross-component development team.

4.3.5 Full-scale rollout

The component-based team setup was used until the end of March, 2013, as the management decided to initiate the full-scale rollout into fully cross-component and cross-functional development teams, as illustrated in Figure 4.5 [45]. The goal of the rollout was to create fully end-to-end capable feature-oriented teams, which could be able to implement any feature from the backlog and deliver it.

In late March, 2013, the team formation was drafted on paper, but the leadership team could not reach a consensus on the issue. However, the full-scale rollout was finally started in April, 2013, and the first three development

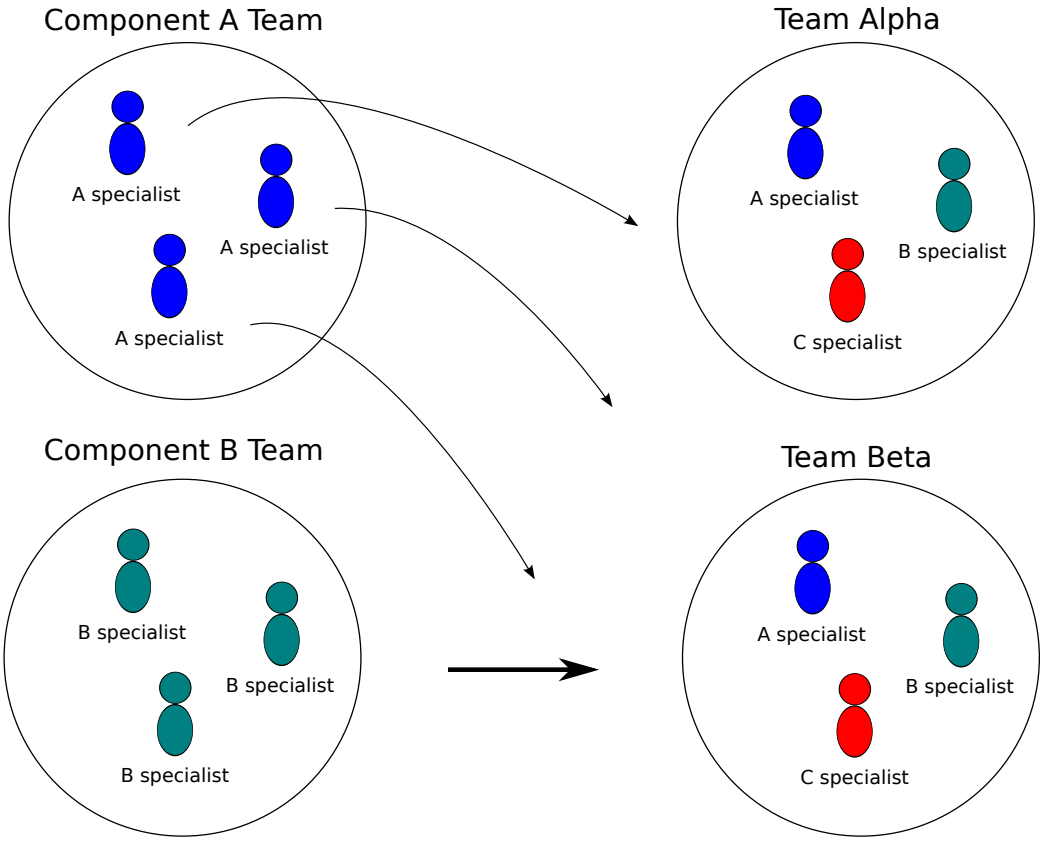

Figure 4.5: From Component Teams to Cross-component Teams

teams were formed. However, due to the ongoing release cycle and the incipient summer vacation period, the other teams were formed from August, 2013, onward.

Team members were allowed to form the new development teams by themselves but the line management had set the frames for the teams. The main principle was that each team should have people with a system, development, testing, and DevOps background. Additionally, teams should be as collocated as possible. The coaches supported teams during the team formation process by working on the guidelines and facilitating team building sessions.

As an outcome of the rollout, two different team setups emerged. People

in sites A and B formed eight teams in total in order to be as collocated as possible. Three of them were fully Agile development teams, another three teams were assigned to implement test automation and CI systems, one team focused on performance issues and the last one focused on a highly complicated component area. At the same time, people in sites C, D, and E formed five teams so that subsystem architects (SSA) and subsystem responsibles (SSR) from sites C and D were mixed with outsourced consultants with component expertise from site E. These five teams were relative large, each of them contained more than 10 people. Some of the SSRs additionally worked as a team coach for their teams.

Working in fully cross-component teams required horizontal implementation competence over all components of the platform in addition to functional competence such as testing and design. However, it was soon realized that this would not be feasible. People had a component responsibility in their own team and they could not contribute in other components because they did not have the required competence. Sometimes one component required much more effort than the other components but there was no one in the team to balance the workload. Therefore, people still tended to work in their own component silos rather than expanding their knowledge over other components.

> *We work a lot with third-party products and, I cannot possibly help, someone else working on another platform. And the other way around. They can't help me, so there's no really a point of having cross-functional teams in that sense. I can't really contribute anything.* — Team member

4.3.6 CI and automation teams

At the time of interviews, the release cycle was eight weeks. More than a half of this was used to verify that the implemented features are working as expected and those do not break anything previously implemented. The

reason for the long verification phase was that the testing and verification were mainly conducted manually as CI or deployment systems did not existed yet. As the system grew, the manual testing became even more complicated and time-consuming.

As a part of the Agile and Lean transformation, it was decided that quality assurance should be invested substantially resources. There were no ways to ensure that the system would withstand the increasing user base or to measure how the system would behave under extremely heavy load.

In August, 2013, three infrastructure teams were established in country Alpha. One of the teams was entirely moved from another organization unit within Ericsson. These teams were assigned to build appropriate CI and test automation systems. The teams were supported by some test consultants located in site D.

The organization had planned to arrange CI workshops on several sites in early 2014. The workshops would be part of the Lean adoption process and people will be acquainted with the subject and its benefits.

4.3.7 The competence pool

After the full-scale rollout, the consultants in site E were divided into five teams together with SSAs and SSRs from sites C and D. However, some of the outsourced consultants had difficulties in adapting to the idea that they should need to contribute other than their core competent areas.

> *In [country Delta] you are very focused on or you're a specialist in one specific area. The feedback we got was that they don't want to work [in fully cross-functional and cross-component agile teams]. That's why we narrowed the teams down into specific flows.* — Team coach

In addition, the consultants were hired only because they are specialists of certain components.

> *Why should they [in country Delta] work with something else? We're not paying them for that. And if we want to have that, then we have to adjust the business relation. They will not come up with architecture improvement proposals either because they're not paid for that. They're just paid to do functionality.*
> — Architect

In the fall 2013, the amount of teams was decreased from five to four and a competence pool was established to improve the utilization of the consultants' capabilities in site E. Some of these consultants were assigned to teams and the rest of them, approximately 20, formed the competence pool as illustrated in Figure 4.6. The competence pool contained people with competence in almost every component area.

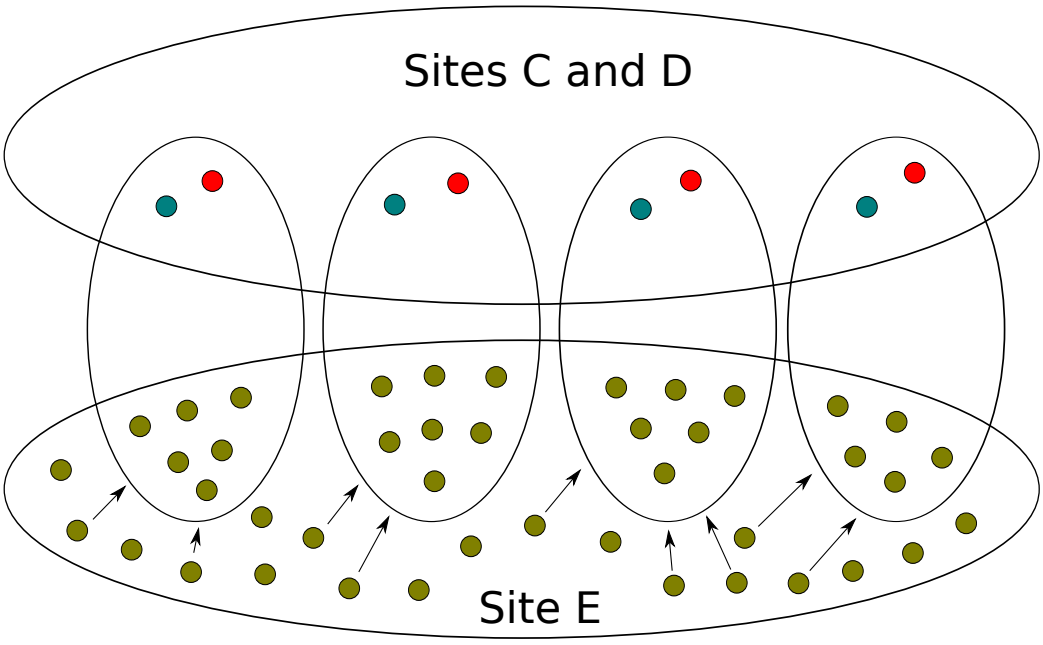

Figure 4.6: Domain Teams with a Competence Pool

At the end of each release, a group of team coaches went through the current situation and redefined resources needed in each team. They could then assign people from the competence pool into their teams depending on the skills and competences needed to implement the features.

4.3.8 Domain teams and a new architecture

Based on the experiences from the fully cross-component and cross-functional teams, the organization moved to the domain-based teams in the late 2013. The product line set up a new high-level architecture that covered all requirements and business areas of the organization. The system was divided into ten domains as presented in Figure 4.7. Four of the domains were functional domains and the rest six were cross-functional domains such as security and system automation, which were involved or included in the functional ones.

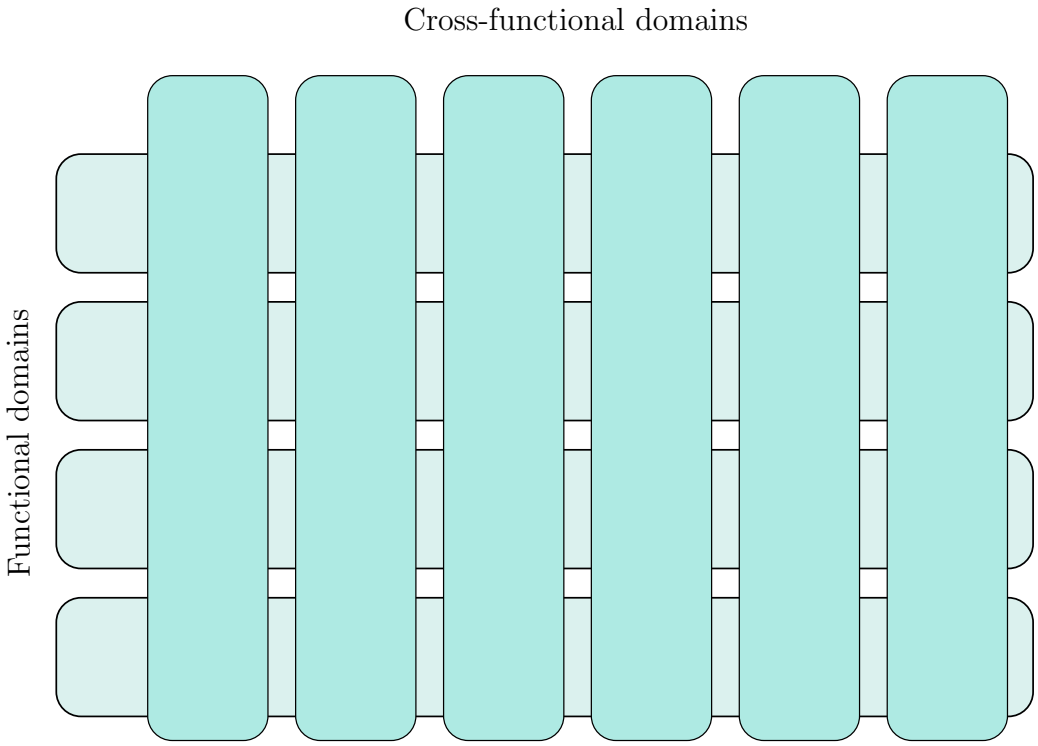

Figure 4.7: Domain structure

Each domain covered closely related components. At the time of interviews, each development team was focusing on a certain functional domain. That allowed teams to focus on a narrowed set of components instead of trying to be capable of implementing features over all components. In addition, teams were readjusted in order to balance the component experts in each team.

> *So we have to make sure that we organize, profile the teams so they can develop, be efficient and say that they succeed with it. So they don't have to think about every type of feature from all the platforms and know that.*
>
> — Product Owner

As the domain team structure was just set up, the idea was to nominate a domain owner from the product line, and a product owner and an architect from the development to each domain. In addition, it was preferred that the product owner and architect of a single domain would be specialized in different component areas in order to increase the cross-functionality within a domain and teams. This should ease the communication and coordination within the organization as teams within a domain should be able to do end-to-end features independently.

4.4 Challenges in the transformation

This section presents the answer to the third research question that was:

RQ3: *What are the impediments encountered during the Agile and Lean transformation?*

The organization had adopted a learning-by-doing approach. Instead of planning all phases accurately beforehand, practices were adopted in small increments, focusing on a few areas at a time. If a chosen practice did not work, it could be rapidly replaced. Therefore, the organization had continuously adapted and adjusted its practices. However, the organization had faced many challenges during the journey, which have been classified into 4 categories, as summarized in Table 4.2.

Table 4.2: Challenges

Dimension	Issue
Management and Organizational	The role of management was changing
	Cultural differences between sites
	Cross-site teams and communication
People	Change resistance
	Lack of Agile training
	Coaches are overworked
Process	Overlapping and vague roles
	Lack of a process definition
	A waterfall driven surrounding organization
	Lack of a common way of coaching
Technical	A high degree of technical debt
	A highly complex platform
	Ill-defined requirements management
	Lack of continuous integration and test automation

4.4.1 Management and organizational

The role of management was changing. The organization was shifting away from the command and control management style and teams were encouraged to take more and more decisions at the team level. However, this had invoked uncertainty especially among some managers who had become accustomed to the traditional management style.

The traditional roles in the organization were also under a change. For instance, there had been many project, product and line managers, whose responsibilities had changed or might change in the near future. Some of them became Product Owners, but not all could have that role. Therefore, several managers had been concerned over their roles.

Cultural differences between sites. People at site E had got used to

working in a traditional organization where they were expecting someone to tell them what to do. This had considerably affected how they work with their cross-site team members. Thus, it had been really hard to get the consultants to be part of the Agile organization.

> *They [at site E] were asked to name a coach to each team, but then also a chief coach was nominated who tells what the other coaches should do.*
>
> — Team member

Furthermore, sometimes it was difficult to get the correct status from the consultants in site E as they tended to say that everything was in order, even if they would have had major problems.

> *They [people at country Delta] don't talk in a same way as we do [people at country Alpha]. If they have problems, they won't say: "Hey, we have problems." They express it in a roundabout way and we didn't always understand it.*
>
> — Team member

Additionally, a turnover rate of the consultants at site E was considerably high, which affected the team work. It was challenging to work with a team where people changed constantly. Even though the consultants had the component knowledge, they might not have had the platform knowledge and it took time to learn it well enough to be efficient at development. This also required extra resources from the system people and technical coaches, who were needed to provide both Agile and system training for the new people.

> *I asked them: "How many of you have been working [in the organization] less than six months?". And I was really surprised when more or less 50 percent of them raised their hands.*
>
> — Coach

Due to the gab between the different cultures and personal preferences of the consultants, the team formation between sites C, D, and E had been done differently in comparison to the other teams. These teams had needed more adjustments and tradeoffs regarding adopted Agile practices.

Cross-site teams and communication. The organization could not avoid cross-site teams because sites had different functional and component competence. At the time of interviews, more than half of the teams were distributed to several sites, which impeded team work and intra-team communication.

Distribution of people within a single team had hindered team spirit as communication was mainly occurred via electronic communication channels. In the distributed teams, some team members had not met each other face to face and did not know each other well. They did not have possibility to have such informal "corridor talk" with the off-site team members as in collocated teams.

> *Q: Do you feel that you are really a team?*
>
> *A: No, I don't. I am a team player and I like working in teams, but I don't feel that we have a team spirit. And, I guess it's hard, when you have multiple sites. As long as you don't know the people, you can't possibly care for them either.* — Team member

In the teams with the competence pool, there could be team members who did not have other team members in the same site as everyone else was in other sites.

> *In my team, we have two members from (site D) and we have one member from (site C). And then we have people in (site E).* — Team coach

> *It's the first time I'm working within a team that, it's only me. I mean that I'm sitting alone and the rest of the team is far away.* — Team coach

> *It's difficult to have a process when it's based on communication and working close to each other and then you're distributed so it's a challenge.* — Coach

Additionally, communication was constantly problematic between outsourced consultants and the other organization. The organization did not have a

proper video connection between site E and other sites because the consultants at site E could not use the same video conferencing system than the other organization. Therefore, they had been forced to use only an audio connection when communicating with people in site E. Furthermore, the quality of the audio was not always good enough.

> *When keeping teleconferencing meetings, a terrific amount of information is always disappeared somewhere. You can't see people, you can't read them, people talk over each other, you can't actually hear anyone anymore. And it actually kills Agile.* — Product owner

4.4.2 People

Change resistance. Some attempts to initiate the transformation existed already in 2012 but the discussions did not lead anywhere as the issue polarized the organization. Some would have wanted to keep working in the old way and those who would have been willing to change, had separated views of how the transformation should be conducted. Additionally, the leadership team comprised people who were not keen to shift to Agile and Lean. Some of them would have wanted to focus on delivering new features to the customers instead of initiating a large-scale change process that would interfere with the whole organization. They had said: "Let's try to change something small, but we cannot make any large-scale changes right now." Due to these issues, the management could not reach a consensus on the transition and the transformation did not progress.

> *The top operative management located in [site C], and they hadn't adopted the Agile philosophy. There was so much resistance that it was absolutely impossible to drive the change from bottom up.* — Team member

However, the organization switched to another organization unit within Ericsson in 2013 and the top management was replaced with people who started

driving the transformation forward. The new management reorganized the leadership team during the summer and fall of 2013 and some were replaced with new leaders with Agile and Lean mindset. This helped to mitigate the change resistance from the management level and to establish a driving force that would lead the transformation process.

At the time of interviews, there were still some people in the organization, who did not care about the new Agile and Lean way of working.

> *In product management there's still some belief that a plan is the truth and try to fulfill that is a good thing.* — Coach

> *Our salesmen have very traditional ways of working (...). They have difficult to see the importance of the Agile and the benefits with it.* — Manager

Lack of Agile training. Approximately, a half of the people in the organization had received Agile training. A few years prior to the acquisition of the platform, Agile training was provided in another organization unit within Ericsson from where several people later moved to this organization unit. However, the other half of the people had never got suitable training regarding Agile practices. The first reason was that the country Beta had had no time to arrange a training session. The other reason was that some of the people in country Alpha had been busy during the time when the training sessions were arranged. Therefore, some had a limited comprehension of working in an Agile way.

> *A: We have skipped it [a retrospective meeting] last, some weeks?*
>
> *Q: Why did you skip it?*
>
> *A: Because we we thought like it is not needed much, because it is needed only whenever a new release is coming up. Like some discussions for improvements (...). But now, we are more focused on delivery.*
>
> — Team member

Additionally, people in the organization lacked a common language when talking about Agile practices. This was partially due to the fact that some people had not got suitable Agile training.

> *Q: What should be improved first?*
>
> *A: First, all terms should be defined. What a feature is and what a story is. And an epic, so that those would be clear and everyone would call things by their proper names.* — Team member

> *Sometimes I send out mail or call people and discuss these, what I feel is basics. And then, for example, I talked to somebody about definition of done. But then, yeah they kind of agreed some of them, but a couple days later you get back some questions, "what is the definition of done". And then you realize, okay. We have to really go back to.. so it's really difficult to do coaching or advice because I don't know where we are. I agree, it's kind of a problem.* — Coach

Coaches were overworked. There had been two coaches from the beginning of the transformation, who had been coaching both the management and the teams. The coaches had been providing insights for the leadership team based on their previous experiences of Agile and Lean transformations and helping them to drive the transformation forward. They had also been working as a team coach to several teams in country Alpha, mainly focusing on the new Agile teams. Additionally, lots of their time had elapsed while traveling between sites and handling the organizational issues. However, the coaches had been overworked due to the wide spectrum of duties. This had lead to the situation that they had not had time to coach teams as much as might have been needed or to actively participate in team meetings in order to observe that the teams work according to Agile principles.

> *There is not enough time to coach teams and spend time with them (teams), as there are so many teams and too few coaches. We are not doing well our role as a team coach.* — Coach

To balance the workload of the coaches, some new team coaches were nominated in country Beta, who, in addition to their other duties, supported and guided their teams regarding Agile practices. Additionally, some part-time organizational coaches were hired in country Beta in the summer of 2013. Their responsibilities included mainly coaching the leadership team and facilitating the Agile and Lean workshops. However, they were not involved in at the team level.

4.4.3 Process

Overlapping and vague roles. New roles had been introduced in the organization since the transformation began. In 2012, roles of a subsystem architect (SSA), a subsystem responsible (SSR) and a product owner (PO) were adopted. In late 2013, the domain ownership was started, which entailed a new role, a domain owner (DO).

However, the roles had been loosely defined and there were equivocal boundaries between several roles. For example, the responsibilities of a SSA, SSR and PO were all overlapping each other. They all had, for instance, participated in the prioritization of the team backlog. Additionally, SSAs and SSRs had the system knowledge that some POs lacked and therefore questions were often asked from SSAs and SSRs instead of POs.

Due to the vague roles, there had occasionally been difficulties in communication and information sharing. It had not always been clear who was responsible for what, from whom to ask information or to whom the information should be shared. Especially the line between SSRs and POs had been problematic as some SSRs did the same work as POs, for example, prioritized the backlog.

Additionally, the product ownership was confusing and ill-defined until the late 2013. There was a huge gab between the product line and POs. The product line talked to the customers and decided what to implement with

the help of system people. POs did not participate in the prioritization of the backlog. In general, prioritized requirements came from the product line and POs were mainly responsible for planning how his or her teams could implement given requirements on time, especially how the requirement should be split and how to prioritize these subparts. Furthermore, POs had been assigned for each component area, but still each Agile team had a dedicated PO.

Another problem relating to roles was a missing role of scrum master (SM) despite the fact that the organization had decided to use Scrum as a development process. Some of the interviewees said that they did no need SMs because Agile coaches were doing that work. For instance, coaches sometimes facilitated team meetings such as a retrospective, but they did not have time to do that in a daily basis.

Lack of a process definition. The development process was not well defined in the organization and the teams had been given a freedom to adopt Agile practices independently. Therefore, people at the different level of the organization and different sites seemed to have a different view of how they were actually working in the organization.

> *We are using Scrum. I wasn't making the decision of using Scrum. If I had, I would have told them that Kanban is the right one for us. We just can't commit ourself to 2-week sprints. Because sometimes new requirements come from a customer, which should have done right away. Even if we are in the middle of a sprint, we just can't say them no.* — Line manager

> *This feels like we are doing things but we don't have any process. But being still very Agile.* — Team member

> *Q: They don't have the process and, a retrospective is part of the process [in country Alpha].*
>
> *A: Not even retrospective? How can you run an Agile organization without, continuous.. learning and improvement. That's interesting. I didn't even*

know that. — Coach

Due to given freedom, the Agile methodologies and practices had not been systematically adopted between teams. The reality was that there were only a few teams that had a 2-week sprint cycle, regular retrospectives, sprint reviews, and sprint planning meetings. Most of the teams had adopted Scrum framework only partially as they had dropped out some of its key elements.

I think Scrum is a very good start, and when you know Scrum then you can shift into other stuff. My feeling here is that we have kind of trying to take a shortcut and doing other stuff immediately, so some of these ground pieces is actually missing in quite many teams. — Coach

A waterfall driven surrounding organization. At the time of interviews, the teams had been seen rather Agile but the surrounding organization was still highly waterfall driven. The release process, for instance, contained several sequential phases, as presented in Figure 4.2. Each release started with a planning and design phase and ended up with a long test, release verification and integration phase.

If a customer says that something doesn't work, it takes a very long time to get it working. It has to go through so many steps in the waterfall before it comes to the development, which then tries to do it in an Agile way. And after that, it goes to the next waterfall, to the release side. — Team member

Lack of a common way of coaching. Coaches at the different sites had not adopted a single way of coaching. The team coaches at country Beta tried to get their teams to use the same practices, whereas the coaches at country Alpha did not drive any particular practices to their teams.

4.4.4 Technical

A high degree of technical debt. One bottleneck that prevented the transformation was a high degree of technical debt in the system. The sys-

tem was originally designed only for a single customer. Additionally, the development in the previous organization had occurred within strict deadlines, which together had resulted in a situation where lots of shortcuts had been taken in the development. Hence, the system was not stable enough to be scaled up for a larger pool of users and system improvements were needed before implementing new features would have been reasonable.

During the 2012, a lot of system improvements were done. In addition, many components were replaced with Ericsson's own components. This was one reason why people were disinclined to start the transformation as it would have interfered with teams' work.

A highly complex platform. The platform comprised several highly complex components that were dependent on each other. If a single feature could not be implemented within a release, it might have affected several other features because of many cross-feature and cross-component dependencies, as illustrated in Figure 4.8. Sometimes this resulted in that some other features could not be released even if those would already have been implemented.

Development of a singe component and the integration of these components required highly skilled and specialized developers. Therefore, it had been really difficult to establish fully cross-component teams that would have been capable of implementing all given features.

> *What we are doing in our reality, I don't think it's possible to be complete Agile in the way of working. (...) and it's much easier to work on one product and be completely Agile, but now we have so many different functional flows which are dependent on each other, and it's very difficult to handle that on an Agile level, I would say.* — Manager

Additionally, learning a single component well in order to be a productive developer could take up for one or two years. People, who started learning the platform in early 2012, seemed to have gained enough knowledge to understand the system so that they could work efficiently in Agile teams and

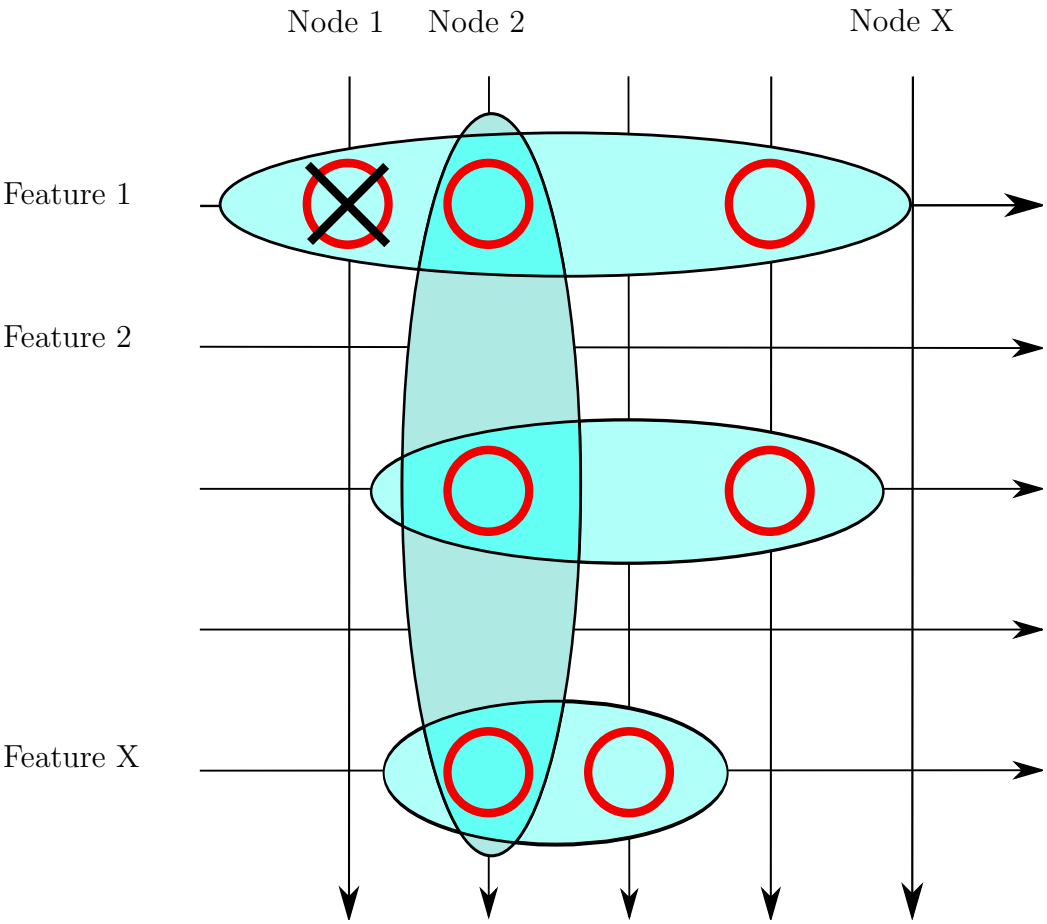

Figure 4.8: A complex platform

implement new features.

Ill-defined requirements management. The features put into the backlog had usually been fairly large. Approximately 15–20 of them could be implemented within a release, whereas the backlog contained approximately 150 features and improvements in total. Therefore, the total lead time of a single item could be several years. During that time, the requirements of the customers might change several times.

> *Q: Is there anything else you think that should be done or improved?*
>
> *A: Breaking down the requirements or items in the backlog into smaller items. (...) I think that is very important, because most of them are too big really,*

> *because they are 500, 1000 hours and it's difficult actually to get the teams to start with them. (...) And the customers do not need to wait until April next year. They get something at least in January.* — Manager

People were generally satisfied that the common backlog had been taken into use. It had been seen as a good way to increase needed transparency in the organization. Additionally, it enabled the organization to start measuring the total lead time of backlog items. However, there were still some concerns regarding it. Some had not been convinced of working with the common backlog. The organization lacked common rules of how to use it, for instance, people had had different views of what they can take from the backlog.

> *We have no a such thing as a common backlog. We have already admitted several times that we have many different backlogs. (...) We actually had problems as someone says that we have a common backlog, but when a team starts going through first items of it, they couldn't understand anything. They just don't have right competence. An item they were able to do was about twentieth on the list and they were told that they aren't allowed to do that yet.* — Product Owner

Additionally, the features in the backlog were usually poorly defined. Neither the teams nor the POs always understood the actual requirements of a feature. Finding the person who knew what was really wanted took time from the developers and then they had even less time to actually implement the feature.

> *When a task has become into the backlog, the requirements aren't yet clear. At that moment, it's known that this is wanted and then you need to go to find out what is really wanted.* — A team member

Lack of continuous integration and test automation. At the time of interviews, the organization had limited capabilities of being Agile as they did not have a continuous integration system. Developers could not deliver

the code frequently into the test environment in order to get instant feedback about the implemented functionality. They were still highly dependent on testers who performed testing manually. Due to the complexity of the system, the testing of all dependencies of the different subsystems was highly demanding and time-consuming when performed manually.

4.5 Success factors in the transformation

This section presents the answer to the fourth research question that was:

RQ4: *What have been the success factors during the Agile and Lean transformation?*

The journey toward an Agile and Lean organization was still in progress and there were still lots of work required. Thus, there were not many success factors yet. However, several positive aspects have been discovered. The findings are presented in the following subsections and summarized in Table 4.3.

Table 4.3: Success factors

Dimension	Aspect
Management and Organizational	Reform of the leadership team
	Organizational culture
People	Liaisons
	Empowered individuals
Process	Communities of practices
	End-to-end flow
Technical	The common backlog

4.5.1 Management and organizational

Reform of the leadership team. The leadership team was reformed in order to mitigate the change resistance from the management side. Several managers were replaced with individuals who were proponents of Agile and Lean, and who had a strong passion to implement Agile and Lean in the organization. This created a strong leading force for the transformation.

Organizational culture. The organization had abandoned the command and control management style. Everyone in the organization, including team members, was allowed and encouraged to contribute and bring up their ideas and opinions regarding the transformation. It was not a single person or the leadership team that would have taken the decisions and said what to do. For instance, the Agile coaches were much involved in the design and execution of the transformation and the management used their knowledge of how things could be made. Additionally, team members were also heard and their wishes regarding ways of working had been inquired.

The management culture was highly permissive. The management had endorsed an organizational culture that enabled an experimental attitude among people without the fear of punishment. Furthermore, the team practices or ways of working were not forced but teams could choose practices that were well suited for their needs. Additionally, a positive atmosphere was perceptible during the value workshops. People were encouraged to have fun and share the passion with each other.

> *We have a very permissive management culture and I'm very satisfied with our line management. We have much freedom to test various ways of doing things. And we are encouraged to have an experimental attitude.*
>
> — Product Owner

4.5.2 People

Liaisons. Having liaisons had been seen as a positive matter. Liaisons were developers from site E, who moved to another site on either country Alpha or Beta for a certain period of time, usually for several months. Their responsibilities during the visit had been to get to know the people in the site, to increase their awareness of the development process, and to communicate this information back to their team members at site E.

> *Q: What was the reason you came here [to site D]?*
>
> *A: I came here to understand the actual requirements from the customers and to find out how we could improve the communication with my teammates so that they would understand better what they are doing. And communication between developers and the actual architects, who are giving the requirements.*
>
> — Team member

Additionally, the visiting developers were also helping the coaches to understand the developers at site E better.

> *It's helping a lot (having a person from site E) here. To understand how the people [in site E] understand things, and how they want things to be done. How they want to be coached. That kind of things.* — Coach

Empowered individuals. Individuals in the teams were taking more responsibility of what they were doing instead of pointing at the management. Additionally, people were more engaged and if they were not able to do something, they did not just stop working but started doing something else.

Team members were allowed to form the new development teams by themselves. The management did not mandate the teams but allowed individuals to discuss together and build the teams based on their best judgment.

4.5.3 Process

Communities of practices. The cooperation and information sharing within component and functional areas had been upheld even after the component team setup was ceased and the cross-functional and cross-component Agile teams were started. Individuals who share a common interest were encouraged to establish a community of practice (CoP) in where they could discuss about the subject. At the time of interviews, at least each component had their own CoP. The CoP meeting could include, among others, discussion about needed improvements or training regarding a component. Additionally, a coaching CoP was started in the late 2013. One of the most important goals for the coaching CoP was to get all coaches together and to establish a common way of coaching between each site.

End-to-end flow. The end-to-end flow had been improved after Agile and Lean were introduced in the organization. The cross-functionality had been improved as most of the development teams had knowledge of the system, development, testing, and deployment. That had eased the teams to develop end-to-end features independently with fewer handovers.

At the time of interviews, three development teams in site A had a DevOps person and two of them actually were at site F. Having DevOps people in teams should improve cooperation with the operations that had earlier been rather separated from the other organization. Additionally, that should help teams to receive information about customer feedback and inquiries that the operations handled at site F.

Additionally, setting up a team, called PO Cloud, encompassing all product owners, a portfolio manager, a test manager and a UX lead enabled to bring people from both the product line and the development closer together. This should help to improve the requirements in the backlog and to provide more accurate information about requirements for the product owners.

4.5.4 Technical

The common backlog. The common backlog improved visibility among all stakeholders. It helped the product line to prioritize features together with the customers, who could then suggest new features and propose how to prioritize them. Additionally, the product line and the customers was able to see the status of each feature implemented by the development teams.

Chapter 5

Discussion and Conclusions

This chapter presents the conclusion of the study. First, the results are summarized in Section 5.1. Second, the results are discussed and compared with the existing literature in Section 5.2. Third, the thesis is concluded in Section 5.3. Fourth, the limitations of the results are presented in Section 5.4. Last, some topics are presented for the future studies in Section 5.5.

5.1 Summary of the results

This section presents answers to the four research questions which were led from the following research problem:

How are Agile and Lean adopted in a large and globally distributed software development organization with a steep growth trajectory?

5.1.1 Motivation for the transformation

RQ1: *Why is the Agile and Lean transformation started in the case organization?*

Based on the studied case project, five main factors were identified that drove the organization toward Agile and Lean adoption. First, Agile and Lean are part of Ericsson's strategy nowadays. Ericsson strives that its software development units would adopt Lean principles and Agile practices. Second, the organization wants to streamline the whole end-to-end flow so that all non value adding procedures would be removed and the lead time of new functionalities would be shorter. Third, the development teams had been working in a component based structure but that had not been an efficient and effective way of working. The organization wants to improve the productivity of the development teams. Lots of development occurred in component silos, which resulted in several handovers and a long cycle-time. Fourth, people in the organization have been distributed to several sites and countries. The spirit between the sites has not been as good as the management would have wanted. Additionally, the gab between different functions has been too large. Last, the previously used a sequential process did not work well nor was it well managed. Many decisions were dictated by the management and requirements changed in the middle of the development.

5.1.2 The transformation approach

RQ2: *What are the key phases in the Agile and Lean transformation?*

The organization started working in a component based team structure in early 2012. However, in the middle of the year, the product ownership was adopted as the system had grown and the amount of dependencies between components increased. Thus, it became essential that someone would actively prioritize new features and watch over different components. At the same

time, the development teams started working in virtual teams. For each new feature, individuals with a specific competence were picked from the component teams and formed a virtual feature team. However, this approach was complex and challenging as not always were right resources available.

In early 2013, the leadership team together with the Agile coaches started planning the new direction for the organization. As an outcome, they decided some next steps that they should take in order to foster the Agile and Lean adoption. One step was establishing a common backlog that would contain all features that the organization would like to deliver to its customers in a prioritized order. The second step was to initiate the value discussion in order to improve the affinity between employees in all sites.

The common backlog was ready in the summer of 2013 with all features and improvements there. Since then, the visibility had been improved because all stakeholders were able to see the status of each feature.

Discourse about moving into fully cross-component and cross-functional Agile teams had already started in early 2013 among the managers and Agile coaches. In March, 2013, a pilot team was established encompassing experienced volunteers from two countries. The team was fully cross-component and cross-functional. Despite the fact that the pilot team would have been highly competent, the experiment was ceased within a few weeks as team's existence disturbed other teams. Hence, less than a month later, the decision of initiating a full-scale rollout into fully cross-component and cross-functional development teams was done. As a result, three new Agile teams were formed during the spring. The rest of the teams were formed during the fall, 2013.

The organization had decided to improve the quality assurance practices. For example, a major part of testing had been conducted manually and barely nothing had been automated. In August, 2013, three infrastructure teams were formed, whose task was to implement appropriate continuous integration and test automation systems. These systems would also enable that developers would get feedback more rapidly. Thus, these systems were

vital to support Agile development.

Resulting in the rollout, experts in sites C and D formed five teams together with outsourced consultants in site E. However, the consultants had not become accustomed to work in cross-component Agile teams. Additionally, they were not much willing to expand their competence over other component areas. Therefore, these teams were rearranged so that the amount of teams was decreased to four and the rest of the consultant formed a competence pool. From the pool, the team leaders could pull resources to balance work load for each release.

The last major reform was establishing a new architecture and forming domain teams, as per the architecture. The organization had become into a conclusion that it would not be realistic to assume that teams could really be fully cross-component as that would require extensive knowledge over several components. Additionally, single team members would still focus too much on their own components without ability to assist with other components. Hence, teams were assigned to certain domain areas, which required knowledge only a few and related components. The domain teams should be more capable of implementing end-to-end features. Additionally, a domain owner was assigned to each domain, which should ease the communication and coordination within each domain.

5.1.3 Challenges in the transformation

RQ3: *What are the impediments encountered during the Agile and Lean transformation?*

The challenges in the transformation were divided into four categories: management and organization, people, process, and technical.

First, some managers found it difficult that the management style would change. The traditional command and control management style was aban-

doned, as development teams were encouraged to be self-organizing and take decisions at the team level while the role of managers would be more collaborative and supportive. Additionally, the cultural differences caused problems. Especially consultants at site E had got accustomed to hierarchical organization structure, so they had difficulties in adapting to the Agile way of working. Then, the organization could not avoid cross-site teams because different sites had different competence.

Second, there were a few people related issues and most of them related to resistance to change. Especially senior managers were not keen to initiate the transformation. The leadership team, for instance, encompassed people who did not have the proper mindset and would have wanted to stay in the old way of working. At the time of interviews, there were still some people who thought that the organization cannot be fully transformed into Agile and Lean and that they are wasting time when conversing about the issue. Additionally, many people in the organization lacked Agile training due to lack of time. Therefore, some teams may not have sufficient understanding of independently adopt Agile practices. Furthermore, Agile coaches had been overworked and they had not had enough time to coach teams with the intensity the coaches would have wished.

Third, several process-related challenges were identified. First of these was that some roles were too vague and overlapping each other. For instance, it was difficult to draw a clear line between subsystem architects, subsystem responsibles, and product owners, who were partially doing the same work. The vague roles also hindered communication within the organization as people did not have clear areas of responsibility. Additionally, the role of a product owner was unclear as they did not participate in the prioritization of new features. Moreover, despite adopting Scrum, they had no scrum masters. The second found process-related challenge was the process itself. The process was loosely defined and teams had freedom to tailor Agile practices by themselves. There was not a clear picture how teams were actually

working in the organization. The third process-related challenge was that the surrounding organization was still highly waterfall driven with several handovers, freezes and gates. This hindered teams' ability to be Agile. The last process-related challenge was related to coaching. Agile coaches lacked a common way of coaching, which made it difficult to build a common base for teams.

Last, a high degree of technical debt partially hindered the transformation process as the organization had to focus on system improvements before putting teams' effort on learning a new way of working. Additionally, the system itself was such a complicated that it required much time to learn it in order to be capable of implementing new features effectively. Then, due to missing continuous integration systems and test automation, teams could not really be Agile as they relied on manual testing and long integration phases.

5.1.4 Success factors in the transformation

RQ4: *What have been the success factors during the Agile and Lean transformation?*

First positive finding was that the organization made an aggressive move to accelerate the transformation by reforming its leadership team. The new leadership team was full of Agile and Lean minded people.

Second positive finding was that the organization culture had changed a lot. At the time of interviews, it was more permissive and people were trusted. In the daily work, teams were allowed to take decisions on their own. Team members had also been attended regarding the transformation process so it was not only the management who decided what should be done but the team members could present their propositions.

Third positive finding was that learning and information sharing had been maintained by establishing communities of practices which enabled people

with a common area of interest to meet and converse together. This might lead to learning new things and to finding out what could be improved regarding that area.

Fourth positive finding was that the organization had chosen a step by step approach to conduct the transformation. This was reasonable because the surrounding organization and the infrastructure were not yet fully capable of supporting Agile and Lean way of working.

Fifth positive finding was that the organization used liaisons from site E to improve communication and collaboration between the outsourced consultants and the rest of the organization.

5.2 Discussion

5.2.1 Motivation for the transformation

Fry and Greene [25] and Hui [36] all justified the need for transformation that the traditional and waterfall driven development is not suitable anymore. Hui [36] further explained that the existing process needed to be optimized. Cloke [14] explained that teams were not able to work efficiently and they needed better ways of working. In the case organization, they had quite similar situation. The used process was rigid and slow. The process also hindered teams capability to implement new features efficiently. Thus, they looked alternative ways to work by adopting Agile and Lean.

5.2.2 The transformation approach

According to the existing literature, there are two main approaches to conduct the transformation: a big bang [25, 53] and stepwise [10, 13, 27, 78]. The studied organization selected the latter one mainly because they thought that it would be sensible to perform the transition using a step by step approach. The organization was not fully ready for full-scale Agile and Lean adoption partially because some supportive functions were missing such as a continuous integration system and people were still learning the acquired system. However, sometimes it seemed that the organization could have progressed a little bit faster and it took too small steps. For example, the way the teams adopted ASD methods was rather blurred. They were using a backlog for user stories but team members were not estimating the working items.

According to several authors [10, 13, 27, 29, 39, 49, 56, 57, 65, 78], Agile transformation was usually started with a pilot project in order to evaluate different Agile methods and practices in small-scale. This was the approach also in the studied case project, which initiated a small pilot team to evaluate how it could work as a cross-component and cross-functional Agile team. However, the experiment was not very successful as it disrupted the other organization. The experiment might have been more successful if the pilot team would have been more like an average team. Now it was full of very talented and experienced people, who were keen to try working in an Agile way. Even if the pilot team would have worked longer, the obtained results might have been distorted because the team was built to be good.

Petersen and Wohlin [60] and Benefield [7] described how the transformation was started at the team level. According to Petersen and Wohlin [60], the organization first set up development teams and a product backlog. After that, new Agile practices were adopted incrementally. According to Benefield [7], Agile practices were first adopted at the team level and after that

scaled up with Lean principles to other organization. In the case organization, the transformation was rolled out at the team level after the pilot phase. Component teams were incrementally turned into Agile teams and different team setups were evaluated. People were switched between teams in order to find a good balance between teams. At the same time with the team rollout, an organization-wide common backlog was built.

5.2.3 Challenges in the transformation

Roles have been a source of conflicts and issues in the case organization. According to the existing literature [80], mixing new and old management roles may lead overhead. That is what had happened in the case organization. There were still traditional project and product managers who worked side by side with product owners. However, the roles had been vaguely defined, which had resulted in problems in day to day work. It was clear that in a large organization in where people have certain roles and responsibilities, it could be challenging to refine roles so that everyone would be satisfied.

The existing literature suggests that the process should be build up and only essential practices are only adopted [9, 78]. This would eliminate unnecessary waste as per Lean principles. The case organization had started to implement practices in small increments, or step by step as they called it. However, the steps could have been larger. For example, it was weird to see that developers were not actually estimating stories even though they were implementing them and they knew what they were capable of. It was said that they first learn how to use the backlog and start estimating backlog items later.

It was confusing that the organization had not adopted the role of scrum master. Approximately, a half of the personnel had never got appropriate training regarding Agile practices due to lack of time and a need to prioritize work. Additionally, the coaches had been overworked and they had not been able to support and coach teams as much as would have been needed. More

intensive coaching would have been important especially during the early phase of the transformation when teams were adopting and learning new practices. Experienced SMs could have helped teams to survive better in an Agile environment and guide team members regarding Agile practices.

5.2.4 Success factors in the transformation

According to the existing literature [2, 10, 21, 25, 50, 78], a management support and commitment are vital to successfully transform the organization. Additionally, it is important to have a buy-in from all stakeholders [55]. In the case organization, the support and strong commitment of the leadership team and top management had had a central role during the transformation. The management was even reformed to improve the support from the management side and to ensure the management buy-in. It helped that the driving force had been the management who believe Agile and Lean way of working. It is easier to transform the whole organization when the management believe in the Agile and Lean.

The organization had strongly focused on building an active Agile and Lean mindset in addition to supportive organizational culture. According to the existing literature [2, 21, 52, 55, 78], these have an major role in the success of the transformation. Without a proper mindset and organizational culture, the people in the organization could not work in an Agile or Lean way.

According to the existing literature [8, 25, 27], a CI environment is vital to support Agile organization. The case organization had invested lots of time and effort in building appropriate CI and test automation systems in order to support teams to work in an Agile way. The systems will provide instant feedback for developers and should allow to decrease total cycle-time of a developed feature.

5.3 Conclusion

To sum up, the main goal of this thesis was to learn how a large, globally distributed and growth telecommunication organization initiated a complex and time-consuming Agile and Lean transformation in one of its software development units. We have particularly focused on the journey of the transformation, including the phases of the transformation in addition to challenges and success factors that have emerged during the journey.

The primary source of information was 32 in-dept semi-structured interviews, which were conducted at four sites in two countries. The secondary sources of information were observations and organizations' internal documents. The research questions were answered based on the data sources.

Thus far the transformation has focused primary on the development teams, leaving the surrounding organization in its old state. The constitution of the new Agile teams has been readjusted several times in order to find a right balance between the teams. Additionally, the development teams have adopted some Agile practices. Some larger changes have already been made at the organizational level, such as adopting a common backlog. However, much time has been using to create an encouraging organization culture that would foster a proper Agile and Lean mindset among people.

The results do not provide best practices but insights on conducting an Agile and Lean transformation in a distributed organization. The organization is on the right track toward its ultimate goal to be a fully Agile and Lean organization. It may be assumed that the transformation gets a new boost when the CI environment is set up and the teams can really work in an Agile way and get instant feedback.

5.4 Limitations

There are several limitations in this study. The major limitation is that only a single organization was studied so the results cannot be generalized. The prospective subjectivity of interviewees is the second limitation. Interviewees were selected by the company's representatives to us. For example, most of the interviewed team members have been actively participating in the change process, so they might have more positive attitude toward Agile than some others. The third limitation is that we were not able to visit site E, which was the site with outsourced consultants. They could have a different perception of the transformation and we could not bring that information out in this study. The fourth limitation is that the questions were not the same for each interviewee. The basic idea staid rather consistent, but the questions were changed a bit over different interviews. The last limitation is that some of the interviewees were conducted in English, which was not a first language of neither the interviewers nor the interviewees.

5.5 Future work

As the transformation in the case company is still in progress, it would be interesting to follow how they can success in Agile and Lean adoption. Based on the latest discussions with the company's representative, they have planned to adopt Scrum in each team and benefits of that could be followed. In addition, building of continuous integration and automation systems will still take a while, so the outcome of that could be good to measure in some way.

Bibliography

[1] Anderson, D. *Kanban*. Blue Hole Press, 2010.

[2] Atlas, A. Accidental adoption: The story of scrum at amazon.com. In *Agile Conference, 2009. AGILE '09.* (Aug 2009), pp. 135–140.

[3] Bass, J. Agile method tailoring in distributed enterprises: Product owner teams. In *Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on* (Aug 2013), pp. 154–163.

[4] Batra, D. Modified agile practices for outsourced software projects. *Commun. ACM 52*, 9 (Sept. 2009), 143–148.

[5] Beck, K., and Andres, C. *Extreme Programming Explained: Embrace Change (2Nd Edition)*. Addison-Wesley Professional, 2004.

[6] Beck, K., Beedle, M., van, B. A., Cockburn, A., and Cunningham, W. The agile manifesto, 2001.

[7] Benefield, G. Rolling out agile in a large enterprise. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual* (Jan 2008), pp. 461–461.

[8] Benefield, R. Seven dimensions of agile maturity in the global enterprise: A case study. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on* (Jan 2010), pp. 1–7.

[9] Boehm, B., and Turner, R. Management challenges to implementing agile processes in traditional development organizations. *Software, IEEE 22*, 5 (Sept 2005), 30–39.

[10] Brown, A. A case study in agile-at-scale delivery. In *Agile Processes in Software Engineering and Extreme Programming*, A. Sillitti, O. Hazzan, E. Bache, and X. Albaladejo, Eds., vol. 77 of *Lecture Notes in Business Information Processing*. Springer Berlin Heidelberg, 2011, pp. 266–281.

[11] Carmel, E. *Global Software Teams: Collaborating Across Borders and Time Zones*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[12] Chau, T., and Maurer, F. Knowledge sharing in agile software teams. In *Logic versus Approximation*, W. Lenski, Ed., vol. 3075 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 173–183.

[13] Chung, M.-W., and Drummond, B. Agile at yahoo! from the trenches. In *Agile Conference, 2009. AGILE '09.* (Aug 2009), pp. 113–118.

[14] Cloke, G. Get your agile freak on! agile adoption at yahoo! music. In *Agile Conference (AGILE), 2007* (Aug 2007), pp. 240–248.

[15] Cockburn, A. *Agile Software Development*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

[16] Cohn, M., and Ford, D. Introducing an agile process to an organization [software development]. *Computer 36*, 6 (2003), 74–78.

[17] Conboy, K. Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research 20*, 3 (2009), 329–354.

[18] Conboy, K., Coyle, S., Wang, X., and Pikkarainen, M. People over process: Key challenges in agile development. *Software, IEEE 28*, 4 (July 2011), 48–57.

[19] Cusumano, M. A., and Selby, R. W. *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People*. The Free Press, New York, NY, USA, 1995.

[20] de Cesare, S., Lycett, M., Macredie, R. D., Patel, C., and Paul, R. Examining perceptions of agility in software development practice. *Commun. ACM 53*, 6 (June 2010), 126–130.

[21] Dorairaj, S., Noble, J., and Allan, G. Agile software development with distributed teams: Senior management support. In *Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on* (Aug 2013), pp. 197–205.

[22] Dyba, T., and Dingsoyr, T. What do we know about agile software development? *Software, IEEE 26*, 5 (Sept 2009), 6–9.

[23] Ebert, C., and Dumke, R. *Software Measurement*. Springer Berlin Heidelberg, 2007.

[24] Ebert, C. E., Abrahamsson, P., and Oza, N. Lean software development. *IEEE Software 29*, 5 (2012), 22–25.

[25] Fry, C., and Greene, S. Large scale agile transformation in an on-demand world. In *Agile Conference (AGILE), 2007* (Aug 2007), pp. 136–142.

[26] Ghanam, Y., Maurer, F., and Abrahamsson, P. Making the leap to a software platform strategy: Issues and challenges. *Inf. Softw. Technol. 54*, 9 (Sept. 2012), 968–984.

[27] Goodman, D., and Elbaz, M. "it's not the pants, it's the people in the pants" learnings from the gap agile transformation what worked, how we did it, and what still puzzles us. In *Agile, 2008. AGILE '08. Conference* (Aug 2008), pp. 112–115.

[28] Greening, D. Enterprise scrum: Scaling scrum to the executive level. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on* (Jan 2010), pp. 1–10.

[29] Hallikainen, M. Experiences on agile seating, facilities and solutions: Multisite environment. In *Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on* (Aug 2011), pp. 119–123.

[30] Hanly, S., Wai, L., Meadows, L., and Leaton, R. Agile coaching in british telecom: making strawberry jam. In *Agile Conference, 2006* (July 2006), pp. 9 pp.–202.

[31] Herbsleb, J., and Moitra, D. Global software development. *Software, IEEE 18*, 2 (Mar 2001), 16–20.

[32] Herbsleb, J. D. Global software engineering: The future of socio-technical coordination. In *2007 Future of Software Engineering* (Washington, DC, USA, 2007), FOSE '07, IEEE Computer Society, pp. 188–198.

[33] Highsmith, J., and Cockburn, A. Agile software development: the business of innovation. *Computer 34*, 9 (2001), 120–127.

[34] Holmström, H., Fitzgerald, B., Ågerfalk, P. J., and Conchúir, E. Ó. Agile practices reduce distance in global software development. *Information Systems Management 23*, 3 (2006), 7–18.

[35] Hossain, E., Babar, M., and young Paik, H. Using scrum in global software development: A systematic literature review. In *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on* (July 2009), pp. 175–184.

[36] Hui, A. Lean change: Enabling agile transformation through lean startup, kotter and kanban: An experience report. In *Agile Conference (AGILE), 2013* (Aug 2013), pp. 169–174.

[37] Iivari, J., and Iivari, N. The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology 53*, 5 (2011), 509 – 520. Special Section on Best Papers from {XP2010}.

[38] Jalali, S., and Wohlin, C. Agile practices in global software engineering - a systematic map. In *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on* (Aug 2010), pp. 45–54.

[39] Javdani Gandomani, T., Zulzalil, H., Abd Ghani, A., Md. Sultan, A., and Sharif, K. Exploring key factors of pilot projects in agile transformation process using a grounded theory study. In *Information and Software Technologies*, T. Skersys, R. Butleris, and R. Butkiene, Eds., vol. 403 of *Communications in Computer and Information Science*. Springer Berlin Heidelberg, 2013, pp. 146–158.

[40] Karkukly, W. When lean meets agile – a complete organization when lean meets agile – a complete organization transformation case study.

[41] Kettunen, P., and Laanti, M. Combining agile software projects and large-scale organizational agility. *Software Process: Improvement and Practice 13*, 2 (2008), 183–193.

[42] Korhonen, K. Evaluating the impact of an agile transformation: a longitudinal case study in a distributed context. *Software Quality Journal 21*, 4 (2013), 599–624.

[43] Larman, C. *Agile and Iterative Development: A Agile and Iterative Development: A Manager's Guide*. Addison-Wesley Professional, 2004.

[44] Larman, C., and Vodde, B. *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum.* Addison-Wesley Professional, 2009.

[45] Larman, C., and Vodde, B. *Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum.* Addison-Wesley Professional, Boston, MA, USA, 2010.

[46] Lee, E. Forming to performing: Transitioning large-scale project into agile. In *Agile, 2008. AGILE '08. Conference* (Aug 2008), pp. 106–111.

[47] Leffingwell, D. *Scaling Software Agility: Best Practices for Large Enterprises.* Addison-Wesley Professional, 2007.

[48] Lindvall, M., Basili, V. R., Boehm, B. W., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L. A., and Zelkowitz, M. V. Empirical findings in agile methods. In *Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods - XP/Agile Universe 2002* (London, UK, UK, 2002), Springer-Verlag, pp. 197–207.

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

[50] Livermore, J. Factors that significantly impact the implementation of an agile software development methodology. *Journal of Software 3*, 4 (2008).

[51] Maglyas, A., Nikula, U., and Smolander, K. Lean solutions to software product management problems. *Software, IEEE 29*, 5 (Sept 2012), 40–46.

[52] Maples, C. Enterprise agile transformation: the two-year wall. In *Agile Conference, 2009. AGILE'09.* (2009), IEEE, pp. 90–95.

[53] Mencke, R. A product manager's guide to surviving the big bang approach to agile transitions. In *Agile, 2008. AGILE '08. Conference* (Aug 2008), pp. 407–412.

[54] Misra, S., Kumar, U., Kumar, V., and Grant, G. The organizational changes required and the challenges involved in adopting agile methodologies in traditional software development organizations. In *Digital Information Management, 2006 1st International Conference on* (Dec 2007), pp. 25–28.

[55] Misra, S. C., Kumar, V., and Kumar, U. Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software 82*, 11 (2009), 1869 – 1890. SI: {TAIC} {PART} 2007 and {MUTATION} 2007.

[56] Nottonson, K., and DeLong, K. Baby steps: Agile transformation at babycenter.com. *IT Professional 10*, 5 (Sept 2008), 59–62.

[57] Paasivaara, M., Heikkila, V., and Lassenius, C. Experiences in scaling the product owner role in large-scale globally distributed scrum. In *Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on* (Aug 2012), pp. 174–178.

[58] Paasivaara, M., Lassenius, C., Heikkila, V., Dikert, K., and Engblom, C. Integrating global sites into the lean and agile transformation at ericsson. In *Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on* (Aug 2013), pp. 134–143.

[59] Patton, M. Q. *Qualitative evaluation and research methods*, 2nd ed. Sage Publications, Newbury Park, Calif., 1990.

[60] Petersen, K., and Wohlin, C. The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Softw. Engg. 15*, 6 (Dec. 2010), 654–693.

[61] Poppendieck, M., and Cusumano, M. Lean software development: A tutorial. *Software, IEEE 29*, 5 (Sept 2012), 26–32.

[62] Poppendieck, M., and Poppendieck, T. *Lean Software Development: An Agile Toolkit*. Addison-Wesley Professional, 2003.

[63] Poppendieck, M., and Poppendieck, T. *Implementing Lean Software Development: From Concept to Cash (The Addison-Wesley Signature Series)*. Addison-Wesley Professional, 2006.

[64] Puleio, M. How not to do agile testing. In *Agile Conference, 2006* (July 2006), pp. 7 pp.–314.

[65] Ramesh, B., Cao, L., Mohan, K., and Xu, P. Can distributed software development be agile? *Commun. ACM 49*, 10 (Oct. 2006), 41–46.

[66] Read, D., Properjohn, G., Aitken, A., and Rosbotham, S. Going agile: A case study. In *19th Australian Software Engineering Conference: ASWEC 2008; Experience Report Proceedings* (2008), Engineers Australia, p. 3.

[67] Rodríguez, P., Mikkonen, K., Kuvaja, P., Oivo, M., and Garbajosa, J. Building lean thinking in a telecom software development organization: Strengths and challenges. In *Proceedings of the 2013 International Conference on Software and System Process* (New York, NY, USA, 2013), ICSSP 2013, ACM, pp. 98–107.

[68] Royce, W. Managing the development of large software systems. vol. Proceedings of the IEEE Westcon. IEEE Computer Society Press, Los Alamitos, CA, 1970, pp. 1–9.

[69] Schatz, B., and Abdelshafi, I. Primavera gets agile: a successful transition to agile development. *Software, IEEE 22*, 3 (May 2005), 36–42.

[70] Schwaber, K., and Beedle, M. *Agile software development with scrum*. Series in agile software development. Prentice Hall, 2002.

[71] Senapathi, M., and Srinivasan, A. Sustained agile usage: A systematic literature review. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering* (New York, NY, USA, 2013), EASE '13, ACM, pp. 119–124.

[72] Sfetsos, P., and Stamelos, I. Empirical studies on quality in agile practices: A systematic literature review. In *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the* (Sept 2010), pp. 44–53.

[73] Shalloway, A., Beaver, G., and Trott, J. R. *Lean-Agile Software Development: Achieving Enterprise Agility*. Addison-Wesley Professional, 2009.

[74] Shaye, S. Transitioning a team to agile test methods. In *Agile, 2008. AGILE '08. Conference* (Aug 2008), pp. 470–477.

[75] Smits, H., and Rilliet, K. Agile experience report: Transition and complexity at cisco voice technology group. In *Proceedings of the 2011 Agile Conference* (Washington, DC, USA, 2011), AGILE '11, IEEE Computer Society, pp. 274–278.

[76] Sutherland, J., Schoonheim, G., and Rijk, M. Fully distributed scrum: Replicating local productivity and quality with offshore teams. In *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on* (Jan 2009), pp. 1–8.

[77] Sutherland, J., Viktorov, A., Blount, J., and Puntikov, N. Distributed scrum: Agile project management with outsourced development teams. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on* (Jan 2007), pp. 274a–274a.

[78] Tian, E. Journey to agility for a large scale telecom system. In *Proceedings of the 2014 International Conference on Software and System Process* (New York, NY, USA, 2014), ICSSP 2014, ACM, pp. 191–192.

[79] Van Waardenburg, G., and Van Vliet, H. When agile meets the enterprise. *Inf. Softw. Technol. 55*, 12 (Dec. 2013), 2154–2171.

[80] Vilkki, K. When agile is not enough. In *Lean Enterprise Software and Systems*, P. Abrahamsson and N. Oza, Eds., vol. 65 of *Lecture Notes in Business Information Processing*. Springer Berlin Heidelberg, 2010, pp. 44–47.

[81] Wang, X., Conboy, K., and Cawley, O. "leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software 85*, 6 (2012), 1287–1299. Special Issue: Agile Development.

[82] Womack, J., and Jones, D. *Lean thinking: banish waste and create wealth in your corporation*. Free Press, 1996.

[83] Yin, R. K. *Case Study Research: Design and Methods*, 4th ed. SAGE Publications, Thousand Oaks, CA, USA, 2009.

Appendix A

Interviews

Occupation	Date	Place	Duration
Line Manager	September 4, 2013	Site A	88 min
Line Manager	October 1, 2013	Site A	74 min
Line Manager	October 7, 2013	Site A	64 min
Coach (x2)	October 7, 2013	Site A	107 min
Developer	October 25, 2013	Site A	73 min
Test Developer	October 25, 2013	Site A	78 min
Product Owner	October 25, 2013	Site A	94 min
Product Owner	October 30, 2013	Site A	74 min
Product Owner	November 20, 2013	Site B	93 min
Developer	November 20, 2013	Site B	81 min
Developer	November 25, 2013	Site A	71 min
Manager	November 26, 2013	Site A	59 min
Coach	November 26, 2013	Site A	31 min
Coach (x3)	November 26, 2013	Site A	26 min
Coach	November 26, 2013	Site A	24 min
Manager	November 26, 2013	Site A	65 min
Manager	November 26, 2013	Site A	30 min

Manager	November 26, 2013	Site A	41 min
Product Owner	November 26, 2013	Site A	28 min
Line Manager	November 26, 2013	Site A	53 min
Developer	December 16, 2013	Site C	51 min
Architect	December 16, 2013	Site C	51 min
Product Owner	December 16, 2013	Site C	27 min
Coach	December 16, 2013	Site C	59 min
Manager	December 16, 2013	Site C	50 min
Coach	December 17, 2013	Site D	59 min
Coach	December 17, 2013	Site D	63 min
Developer	December 17, 2013	Site D	41 min
Consultant (x2)	December 17, 2013	Site D	38 min
Domain Owner	December 17, 2013	Site D	29 min
Manager	December 17, 2013	Site D	48 min
Coach	February 28, 2014	Site A	58 min

Table A.1: Interviews

Appendix B

Observations

Date	Place	Duration	Observed Sessions
October 15, 2013	Site A	7h 5min	Sprint review, retrospective, sprint planning, PO cloud, weekly demo
November 25–26, 2013	Site D	6h + 3h	Value workshop
December 17, 2013	Site C	31min	Weekly demo
January 20–21, 2014	Site D	6h + 3h	Value workshop

Appendix C

Interview templates

General topics and questions

1. Interviewee
   (a) Background
   (b) Role and tasks in the organization
   (c) When did you start in your current role / in the organization?
   (d) What are your expectations toward our research?
2. Overview
   (a) Your history in the organization?
   (b) What's the history of the organization?
   (c) How do you see the current situation of the organization? How are you doing?
   (d) How has the transformation process started to proceed in your opinion?
   (e) What are the reasons for the transformation?

(f) What is the direction the organization should go in your opinion? What's the goal?

(g) When did you start talking about agile and lean in the organization?

(h) What kind of steps have you taken regarding agile and lean?

(i) What are you doing well in the organization?

(j) What could you do better in the organization?

(k) What should you change in the organization?

(l) Change resistance

3. Role

(a) How would you describe your daily work?

(b) How are Ericsson's values visible in your work?

(c) What are the difficulties of working in an agile organization?

(d) What are the most challenging issues in agile transformation from your viewpoint?

4. Agile methods

(a) How have individuals (you) understood the agile principles?

(b) Do you follow these principles?

(c) What agile methods are you using (in your team)?

(d) What is your personal opinion about agile?

(e) What specific pain points have you identified in the agile development process?

(f) How is lean visible in your work?

(g) How is agile seen in the organization?

5. Agile practices
    (a) What Agile practices are working well?
    (b) How do you follow the progress during a sprint / release?
    (c) What do you think about product ownership?
        i. How do you decide what to do in a sprint?
        ii. How do you get the information about needed features?
        iii. Do you have a clear picture of what you need to do?
        iv. Where do you get the information about what to do?
    (d) Continuous improvement?
        i. What do you do to keep up continuous improvement?
        ii. How are you supported to enhance learning?
    (e) What do you do to remove emerged impediments?
    (f) How is the architecture designed?
6. Transition to agile
    (a) How has the transformation affected your work?
    (b) What benefits have you gained from agile methods?
    (c) Have there been any drawbacks regarding agile?
7. Training
    (a) What kind of training have you got relating to your current role?
    (b) What kind of training have you got relating currently used agile methods?
8. Release
    (a) What are the release practices?

9. Communication and collaboration
    (a) How do you maintain informal communication?
    (b) How do you collaborate with other teams?
    (c) Interaction with other people in the company?
    (d) Knowledge sharing
    (e) Daily scrums
        i. Do you use daily scrums?
        ii. How they help you to start your day?
        iii. How they affect you to focus on your work?
    (f) Scrum of scrum
        i. What do you think about SoS?
        ii. Is it a good concept?
    (g) Communities of practices
        i. What CoPs are you participated in?
        ii. What do you think about CoPs?
        iii. Benefits? Drawbacks?
        iv. How could you improve CoPs?
    (h) Off-site collaboration
        i. What is your opinion about the collaboration with other sites?
        ii. What are the challenges in that?
10. Distribution of work
    (a) How is the work divided among sites?
    (b) Team building

(c) Empowerment and motivation in other sites

(d) How is the team culture built and maintained in distributed teams?

(e) How is the work organized among different sites?

(f) Do you collaborate with another sites?

11. Testing

(a) What is the current level of a test environment?

(b) Goals for continuous integration?

(c) What complementary practices do you have to CI?

12. Challenges

(a) What are the biggest challenges at the moment?

13. Needs?

14. Plans for the future

(a) Are there any plans for the future?

(b) What are the issues that should be taken into account when planning the future in your opinion?

(c) What is going to be improved?

(d) Possible stumbling blocks?

15. End

(a) Is there anything else you would like to comment or say?

Role specific topics and questions

Developers

1. Transition to agile

   (a) What are the most challenging issues in agile transformation from developer's viewpoint?

2. Agile practices
   (a) What agile practices are your team using?
   (b) What do you think about agile coaching? Do you get enough support from them?
3. Responsibilities
   (a) What are your responsibilities as a developer?
   (b) Are people taking responsibility in your team for team's results?
4. Agile team
   (a) How would you describe your team?
   (b) Team structure?
   (c) Self-organizing teams?
   (d) What do you think about working in an open environment?
   (e) How have you built the trust among team members?
5. Meetings
   (a) What do you think about agile meetings?
   (b) How often do you have meetings? What meetings?
   (c) Are you identifying improvements during the meetings?
   (d) Do you know how the work is progressing after the meeting?
   (e) Do you have a clear picture what other teams have been doing?

Product owners

1. The role of a product owner

(a) What are the difficulties of being a product owner?

(b) Are there any complaints on your role?

(c) Interaction with other people in the company?

(d) How do you collaborate with teams?

(e) How do you support the work of teams?

(f) How much time do you spend with teams? How often are you available for teams?

(g) What is your role as a product owner for remote teams?

2. Feature handling

(a) How do you choose what to do in each release and sprint?

(b) What is the flow of requirements?

(c) How do you interact with customers or users?

(d) How do you interact with product line?

(e) Working with backlog?

(f) How do you share the vision of features being developed to the team?

(g) How do you share the vision of features being developed to other product owners?

(h) How requirements are documented?

(i) Common repository between sites?

(j) How do you ensure everyone gets the needed information?

3. Backlog management

(a) What is the policy for the backlog management?

(b) What backlogs are in use?

(c) How have teams learned to use backlogs?

(d) How do teams select what to take from a backlog?

(e) Are backlogs available for everyone?

Coaches

1. The role of an agile coach
   (a) What are the difficulties of being an agile coach?
   (b) What are the most challenging issues in agile transformation from coach's viewpoint?
2. The role of SSR
   (a) What does SSR mean in practice?
3. Agile methods
   (a) Are teams using text-book Scrum or have you modified Scrum practices to fit better to teams' needs?
   (b) Are teams allowed to select frameworks they use (e.g., between Scrum and Kanban)?
4. Coaching teams
   (a) Team formation practices (kick-off)
   (b) How do you support teams as a coach?
   (c) How much time do you spend with teams?
   (d) How much help teams ask from coaches?
   (e) How do you promote learning, innovation, and self-organizing?
   (f) How do you motivate teams?
   (g) Do you spend time in other sites?
      i. How much?

ii. How often?

iii. What do you do there?

(h) Meetings / training

5. Coaching product owners

   (a) How do you support product owners as a coach?

   (b) How much time do you spend with product owners?

   (c) How much help product owners ask from you?

   (d) What is the role of product owner in the organization?

6. Organizational coaching

   (a) What do you personally do to build a uniform agile the organization?

   (b) What are the challenges in adopting organization wide agile?

7. Communication practices

   (a) Do you have frequent meetings with other coaches? What are these conversations about?

Architects

1. Agile practices

   (a) How do you do system integration?

   (b) How do you participate in testing?

   (c) How do you collaborate with development and testing teams?

   (d) How is architecture seen in the organization? How much effort is used to it?

   (e) Agile says that architecture kind of emerges during the development process. How is that in practice?

(f) Do you spend time in other sites?

i. How much?

ii. How often?

iii. What do you do there?

2. System core team

(a) What is the system core team? What do you do?

(b) What are its responsibilities?

Product managers

1. Backlog

(a) How do you decide what to include in a release?

(b) How do you prioritize the backlog?

(c) Who makes the decisions of the content of backlogs?

(d) How are the requirements handled?

(e) What is your relationship with the product owners?

(f) What is the division of responsibilities between product manager and product owner?

(g) How do you collaborate with product owners?

(h) How would you describe product owners' role in the organization?

(i) How are you going to decrease the gab between product management and POs?

2. Release practices

(a) Do you have a separate release management team? How would you describe that team?

(b) How would you describe the release management process in the organization?

(c) How long would it take to deploy a change that involved just one line of code?

(d) What kind of release practices have you adopted?

(e) How are releases monitored?

Managers

1. Transformation

   (a) Why didn't you implement agile methods purely as described in books?

   (b) What are the most important factors in the transformation?

   (c) What will happen in the future?

   (d) Have you defined a roadmap for the transformation? Steps?