Bachelor's thesis

2017

Ville Kumpulainen

Aalto University
School of Science
Information Networks program
Information Networks Major

Ville Kumpulainen

Adopting agile software development in large organizations

Bachelor's thesis

15.9.2017

Supervisor:
Maria Paasivaara
maria.paasivaara@aalto.fi
+358 50 3016 847

AALTO YLIOPISTO
PERUSTIETEIDEN KORKEAKOULU
PL 11000
00076 AALTO

BACHELOR'S THESIS
ABSTRACT

Author: Ville Kumpulainen
Title: Adopting agile software development in large organizations
Program: Information Networks
Date: 15.9.2017
Page count: 21
Responsible teacher: Miia Jaatinen
Supervisor: Maria Paasivaara
Language: English

Abstract:

Agile methods have increased their popularity among software development companies, but adopting them in large organizations has not been and still is not easy nor straightforward. I conducted a systematic literature review on why large organizations adopt lean and agile software development methodologies and how these transformations proceed. The keyword search found 1182 unique papers on the subject. Using a specific inclusion criteria, 6 of these papers were included in the final analysis. These papers represented the highest quality regarding their informativeness and narrative of the agile transformation process. The analysis of these papers revealed four major categories of reasons for the agile transformation: Business, process, management and organizational reasons. The most significant reasons were excess process overhead and lack of collaboration inside the organization. Most transformations started from an executive management decision and proceeded in a planned, step-wise manner.

Keywords: Agile software development, large-scale agile, organizational transformation, adopting agile software development

Contents

Abbreviations

ASD	Agile Software Development
XP	Extreme programming

1 Introduction

Agile software development has become an appealing alternative for large organizations striving to improve their performance. Even though agile methods were originally designed for small teams, their benefits have made them attractive for larger organizations as well. The transformation from traditional waterfall model to agile methods in large organizations is challenging, as there is no universal framework that could be directly applied with guaranteed results. One significant challenge is that agile software development methods originally concerned only software development teams. However, in large organizations software development is rarely conducted in isolation. On the contrary, it is typically done in cooperation with for example design, marketing and human resources departments. Since these organizations have their own processes that typically do not follow agile methods, large scale agile transformation must take into account all stakeholders' needs.

The purpose of this paper is to observe why large software development organizations initiate agile transformation and how these transformations proceed. The paper is partly based on Dikert's, Paasivaara's and Lassenius' paper *Challenges and success factors for large-scale agile transformations: A systematic literature review* [5]. This study was based on articles, experience reports and other papers published before 2014. They found that there are not many studies regarding large scale agile transformations, and most of their referenced articles were experience reports. Thus, this research studies articles and papers published after 2014 and attempts to create a comprehensive review of new research data regarding the subject.

2 Background

In this section I first introduce the philosophy behind agile software developemnt. Then I will go through some existing studies, creating an overview of what has been researched so far. Finally I will define large scale agile in the context of this study.

2.1 Agile software development

Agile software development is a collection of methods describing an alternative for traditional software development methods. The fundamental idea is to build software iteratively in small increments while constantly adapting the software according to customer feedback. Feedback is acquired by delivering software to the customer already in the early phases of development. With constant customer involvement changes can be made rapidly even late in development. These principles were explained in the "agile manifesto" [2].

According to Mohammed and Abushama [7], some of the most popular agile methods in the industry are extreme programming and Scrum. Scrum is a software development process for small teams that focuses on project management. A scrum team develops a software product in short sprints, which last from 1 to 4 weeks. In each sprint, the team implements, tests, reviews and ships a finished piece of software

[8]. A sprint is strictly timeboxed and the end date typically does not change. If the team cannot finish what they originally planned to during the sprint, the delivered functionality can be reduced, but the length of the sprint always remains the same.

Extreme programming (XP) is a collection of practices that emphasize on concurrent planning, implementing, testing and analyzing [1]. The aim is to enable efficient incremental and iterative development using eġċontinuous feedback from customer, test driven development and continuous integration. Other important practices include pair programming and constant refactoring [1].

2.2 Adopting agile methods in large organizations

Agile software development methods were originally developed for small software development teams. However, their proved success has made it an attractive alternative for larger organizations as well. Nevertheless, it has been noticed that adopting agile methods in large organizations is more difficult [10]. One significant factor that makes the adoption more challenging is that in small organizations software development teams are typically more independent. Large software teams have more dependencies between other organizations and departmens, increasing the need for formal documentation and communication [11]. Agile methods are largely based on quick, informal communication between team members, such as daily scrums. Once organization and team sizes grow large enough, informal communication is not sufficient to keep all stakeholders up to date with the development progress [4, 11]

Agile software development also causes issues with business processes. In traditional software development the phases of the project are easy to define. This makes setting up milestones, measuring progress and writing contracts easier, as everything can be decided beforehand. With agile development, tracking progress can be more challenging since the requirements and planned features might change throughout the project. This also makes pricing and effort estimation more difficult, which makes traditional software development processes more attractive from management point of view [3].

2.3 Definition of large scale agile

Since this study will target specifically large software development organizations, the definition of large organization will be set to 50 persons or 6 separate teams. This definition is based on Dikert's, Paasivaara's and Lassenius' research [5], in which they studied the success factors and challenges in adopting agile methods in large organizations. In these organizations all personnel do not need to be software developers or engineers. The number may include other positions, such as architects and scrum masters who still participate in the development process. Another important inclusion criterion for the organizations is that they must have been using traditional software development methods in the past, and transitioned to agile methods. The aim of this study is to study specifically organizations that are transforming their processes from waterfall to agile, not merely scaling up agile methods.

3 Research Method

This section covers the research approach. First, the research questions are presented, followed by a description of the research process and further details regarding the research methods.

3.1 Research questions

The purpose of this systematic literature review is to answer the following research questions.

- RQ1: Why do large software development organizations initiate agile transformation?
- RQ2: How do large-scale agile transformations proceed?

3.2 Research process

This systematic literature review consisted of four separate stages.

1. Identification of potential sources
2. Filtering of relevant sources
3. Coding
4. Analysis of coding and aggregation

The data selection process can be split into two different phases. First the selected eletronical databases were searched using predefined keywords. Then the search results were combined and duplicates were removed. In the second phase the results were filtered to exclude irrelevant articles. This filtering was done based on the abstracts of the articles. Once the primary data sources were identified after the filtering, data extraction was done by qualitative coding of the studies. In the final phase these results were analyzed and aggregated.

3.2.1 Inclusion criteria

Due to the research questions and the primary focus of this study, there are four different aspects to guide the inclusion and exclusion criteria: *agile software development, organizational transformation, large–scale and empirical*. Table I lists these aspects and examples of relevant and non–relevant topics related to each aspect.

Table I – Inclusion criteria

Aspect	**Relevant topics**	**Non–relevant topics**
Agile software development	An organization develops software; introduced method is agile	Agile manufacturing; Scrum in management
Organizational transformation	Insight about the transformation process	Before and after comparison; how agile is used in large scale
large–scale	Organization consists of at least 50 people or 6 teams	Scaling up from small–scale agile; single agile team in otherwise large organization
empirical	Experience reports, case studies etc.	Student experiments, theory papers, textbooks

Agile software development as an aspect covers studies focusing on software development organizations that apply agile methodologies in their work. This immediately excludes all other applications of agile methods than software development.

Organizational transformation asserts that primary studies are required to provide insight on organizational transformation, specifically concerning the research questions, iėḣow and why organizations migrate from traditional processes to agile methods. This excludes studies that eġċompare traditional and agile methods or describe how agile methods are used in an organization but not covering the introduction of the methods. Studies that do not explicitly present how the large–scale agile transformation proceeds are not included in this review.

The third aspect states that included studies must conform to the previously set limits. Some papers do not clearly state the size of the software development organization. Merely stating that the organization is large but not stating the actual size resulted in exclusion of the paper. Moreover, adopting large–scale agile methods has to include the whole organization adopting the methods. If only one team or some part of the organization, but not the whole organization, adopted agile methodologies in their work, the paper was excluded.

The final aspect excludes hypothetical and theoretical models and papers. Included studies must present real world cases. Textbooks and theoretical models concerning large–scale agile software development were excluded from this study, as the goal is to observe actual experiences. Also, studies that only report the advantages or limitations of adopting large–scale agile were excluded, since this paper studies the transformation process itself, not the results.

Finally, primary studies must include at least some discussion on all aforementioned aspects. However, even if they contain some other nonrelevant discussion they can still be included, and the relevant parts of the study will be used.

3.2.2 Identification of primary studies

The searches were performed on four different online databases as described in table II. This study only focuses on most recent studies to limit the amount of data, as the scope of this study is limited. Therefore, only studies from 2013 onwards have been included in the searches.

Table II – Databases

Database	**URL**	**# of matches**
IEEExplore	http://ieeexplore.ieee.org	471
ACM	http://dl.acm.org	153
Scopus	http://www.scopus.com/home.url	844
Web of Knowledge	htttp.apps.webofknowledge.com	615 results

The search strings were constructed using boolean operators and aspects presented section 3.2.1, as demonstrated in table III. All databases used in this study supported complex boolean-based search strings, which greatly increased the accuracy of the searches. Preliminary test searches with trivial keywords (such as "agile software development" and "large scale agile") proved that among the interesting articles there is also a vast amount of uninteresting papers. Being able to filter some of those at the search phase reduced the amount of manual labour in the next step. However, the preliminary searches also helped to identify that some actually interesting papers were not left out of the results by accident due to too complex search string.

Table III – Aspects and related search terms

Database	**Keywords**
Agile methods	agile, scrum,"extreme programmin", waterfall,"plan–drive", RUP
Organizational transformation	transform*, transiti*, migrat*, journey, adopt*, deploy, introduc*,"roll–ou", rollout
Only software related articles	(software OR (conference="agile, xp, icgse, ics")) AND NOT (title+abs="manufacturin" OR conference="agile manufacturin")

3.2.3 Study selection

The study selection process was composed of two phases. In the first phase, all studies that matched the search queries were filtered by their abstracts. Furthermore, as the purpose of this paper is to supplement the findings by Dikert et al [12], all studies that were already covered by aforementioned researchers were filtered out of this study. At this point, 49 articles conformed to the set of inclusion criteria set previously. However, not all abstracts were detailed enough to prove that the article covered all required aspects. These cases were included to the full text filtering to make sure all possibly relevant articles are not left out.

The full text filtering resulted in the additional exclusion of 40 articles that failed to comply to the inclusion criteria. The most common reason to this was the organization in question not being large enough as specified in section 2.3. The second most common reason for exclusion was the failure to provide empirical data of the relevant topics. Some experience reports that seemed relevant according to the abstract covered for example solely the results of the transformation and not the process.

3.2.4 Coding of primary studies

The nine primary sources were coded using the Atlas.ti qualitative data analysis software. The coding process followed similar principles as Dikert's, Paasivaara's and Lassenius' systematic literary review of the challenges and success factors for large-scale agile transformations [5]. Thus, five of the seven code families used in the aforementioned study were used in the coding of these studies. The two left out families, *challenges* and *success factors* were not considered relevant for this study, since the reserch questions and goals are different. A description of the code families as well as examples ofo certain codes are show in Table 4. The table also presents the total number of codes and quotations created. A single quotation may contain several codes and belong to multiple code families, which is why the total number of quotation is less than the sum of quotations in each family.

Table IV – Code families, codes and quotations

Code family	**Description**	**Examples**	**Codes**	**Quotations**
RQ1: Reasons to change	Reasons to start the transformation	reducing time-to-market	12	30
RQ2: Transformation process	Statements describing the transformation process	top-down, big bang, step-wise	4	16
Practices	Practices used or established during the transformation	piloting, coaching, continuous integration	4	11
Investing in change	Factors presenting how the organization is investing in the transformation	training, consultants, tools	4	12
Contextual	Contextual codes defined in table 6	agile method, organization size, large-scale definition	7	35

4 Results

This section covers the findings. First, I present an overview of the primary studies. After that I briefly introduce the organizations covered in this research. Finally I present my findings regarding the research questions.

4.1 Overview of studies

In this section I present an overview of the studies, the different organizations and the agile methods applied in different organizations.

Table – Contextual information from primary studies

Study	Company name	SWD org size	Initial state	Agile method	Transformation type
1	Not stated	50 engineers	Plan–driven	Scrum	top–down
2	Not stated	84 engineers	Waterfall	Scrum	bottom-up
3	Ericsson	Not stated exactly	Waterfall	Scrum + Kanban	top-down
4	Samsung	100+	Waterfall	Scrum of Scrums	top-down
5	Not stated	Not stated exactly	Waterfall	Scrum / Kanban	top-down, big-bang
6	Not stated	Not stated exactly	Waterfall	Scrum / XP	top-down
7	2 anon companies	200–400	Waterfall	Lean + agile	top-down
8	Cisco	200+	Waterfall	Scrum	top-down
9	LexisNexis	Not stated exactly	Plan-driven	Kanban + Scrum	top-down

4.1.1 Organizations

Organization sizes. The size of the ten studied organizations varied from 50 to several thousand engineers. Not all papers explicitly reported the exact size of their software development organization, but the size could be concluded as large. This was the case with for example Ericsson [P3], Samsung electronics [P4] and the anonymous company referred to as "ORG" by Roman et al [P5].

Business areas. Business areas ranged from public IT organization [P2] to telecom [P3, P6] and online services for consumers and businesses [P1, P4, P5]. The last mentioned area includes software as a service solutions for businesses, customized software delivery for businesses and online services for both consumers and businesses.

Table V – Business areas

Business area	Organizations
Telecommunications	3
Online services for consumers and businesses	2
IT Services	2
Government	1
Financial services	1

4.1.2 Agile methods applied

All six case organizations applied at least some variant of Scrum in their new software development processes. One organization reported following mainly Scrum but complementing it with elements of Lean principles in their software product development organization [P3]. One team in the same organization reported that they utilized mostly Kanban, but combined it with some practices from Scrum, since especially the retrospective was found essential to their way of working [P3].

The largest software organization studied in this paper (Samsung Electronics) adopted Scrum of Scrums to coordinate the whole software organization [P4]. Additionally, they applied practices such as continuous deployment to support the Lean and Agile way of working. However, inside the single Scrum teams they made use of single, separate agile practices they found useful, such as pair-programming [P4].

Most organizations did not settle for using strictly Scrum. In five out of six cases a combination of multiple agile methods was in place. Nevertheless, all these cases included elements of Scrum. One organization adopted a practice they labeled as "Scrumban", in which they adopted the Kanban way of working, but included the concept of sprint [P5]. Another example of using several methodologies side by side was the large Telecom business [P6]. They adopted Scrum for their information system and information technology development projects, but XP for internal software development.

Table VI – Contextual information from primary studies

Contextual code	Description
Agile method	Agile methods that organizations started using
Business area	Area of business in which the organization operates
Geographical location	Where the organization is located
Large scale definition	What the paper regarded as large scale software development
Multisite / GSD	Mentioning of a multisite organization
Organization size	Mentioning the size of the organization
Research process	The paper defines a distinct research process

4.2 Reasons to change

This section answers to the research question 1: This section answers to the research question 1: *Why do large software development organization initiate agile transformation?*

All organizations in primary studies reported some rationale for initiating the agile transformation. Reported reasons for transformations are reported in table 6

Table VII – Reasons for initiating agile transformation

Reason	Primary sources
Business reasons	
Accomodating change needed	P2, P3, P5
Demand for faster delivery	P1, P2, P3, P6
Need to increase innovation	P5
Remain competitive	P1, P3, P5, P6
Reducing time to market	P2, P3, P5, P6
Process reasons	
Late integration, testing and feedback	P1, P2, P5
Old process does not scale up	P1, P5, P6
Process overhead	P1, P2, P3, P4, P6
Management reasons	
Project management challenges	P2, P5, P6
Schedule and estimation challenges	P6
Organizational reasons	
Lack of collaboration due to organizational silos	P2, P3, P4, P5, P6
Lack of customer collaboration	P2, P5, P6

The top reason for transformations was process overhead, followed by lack of interorganizational collaboration. Traditional waterfall model for software development is a heavy process, which was taking its toll on the efficiency of many organizations. Different teams that were building the same software were not communicating with each other in any other ways than through the formal documentation that was created during the software development. This makes accomodating to change very slow and the whole process becomes very rigid.

4.2.1 Business reasons

One of the most typical reasons for initiating an agile transformation had to do with the business case.

Accomodating change needed. A major issue for several organizations was that they were unable to respond to change requests quickly enough [P2, P3, P5]. One commented this issue reported: *it became clear that we needed to do a proactive change in order to more flexibly react to customer wishes.* [P3]

Demand for faster delivery. Several companies also reported pressure to deliver working software faster [P1, P2, P3, P6]. The pressure was especially strong in companies building primarily custom software for external customers. Smaller companies that had implemented agile methods to their processes were fulfilling customers' needs better. [P6] Wells et al stated: *For this technology-intensive company the challenge of being able to compete in speed to market was achieved through the creation of a culture and mind-set ready to respond rapidly to change, external needs and technological developments.*

Need to increase innovation. One organization also reported the motivation for agile transformation to come from the need to work more innovatively: *We seek agility as a 'driving force' to innovation.* [P5].

Remain competitive. This was one of the most common reasons for agile transformations. Organizations needed to undergo fundamental changes to their software development processes in order to achive the improvements they were looking for. Constant process improvements for their waterfall process had made further improving difficult, as reported by Anwar et al: *This accumulation of tailoring rules has caused what we call "overall process corrosion" – a process that works but is very hard to evolve.* [P1]

Reducing time to market. Long iterations and lead times have become a major issue for companies delivering customized software to external customers. The long time window between the customer requesting a piece of software and receiving it was concidered a major motivation for the transformation. [P5]

4.2.2 Process reasons

Late integration, testing and feedback. The nature of waterfall development made the testing and feedback cycles very long. Anwar et al concidered reducing the total testing effort one of the most significant reasons for improvement. Especially late user acceptance testing had caused major budget overruns in several projects. [P1]

Old process does not scale up. Constant improvements and process tailoring had helped to create a relatively well working and mature development model in some cases [P1]. However, while the old process was working well for old customers, it did not work as well with new projects. Wells et al had reported the heavy emphasis on upfront estimation and planning to make the waterfall model monolithic and heavy, leading to inefficiencies. Having to go all the way back to the beginning of the process to introduce changes was not feasible in most projects. [P6]

Process overhead. Heavy and well-defined process works well in system or safety critical software development but not as well in novel or smaller scale projects. Excessive process overhead had caused software companies to "over-architect rather than doing the job". [P6] Moreover, releaving the rigidity in plan-driven development was a major reason to initiate the agile transformation at Samsung electronics as well [P4].

4.2.3 Management reasons

Project management challenges. Half of the organizations reported difficulties in project management to be a factor in the decision to initiate agile transformation [P2, P5, P6]. Some organizations complained about the increased bureaucratic burdens [P6] and lack of process visibility due to teams working in isolation [P2]. Wells et al also stated that the lack of visiblity and communcation led to teams not being able to deliver specified functionality on time. Agile methods were implemented to emphasize

people and communication over processes [P6].

Schedule and estimation challenges. Heavy upfront planning had also made project scheduling and estimation difficult. In most novel projects customers did not actually know what they wanted, and change requests caused major complications later in the project. [P6]

4.2.4 Organizational reasons

Lack of collaboration due to organizational silos. All but one study reported issues with inter-organizational communication. Ayed et al reported that *"business stakeholders do not collaborate enough with technical team members"* [P2]. Roman et al stated the lack of collaboration to be one of the most significant reasons for the transformation: *"The transformation to agile also aims to reduce the communication gaps between business and IT"* [P4]. Kim et al reported that Samsung Electronics organized "cross-functional teams to reduce the silo effect from component based team model". It was found that in most projects it is not enough that the business stakeholders deliver a list of required functionality and the technical team implements them. In reality, the whole design and implementation project requires constant collaboration. [P2, P6]

Lack of customer collaboration. Missing customer collaboration was also reported as a singificant factor in replacing waterfall development. Weak customer engagement resulted in poor customer experience [P2, P6]. Roman et al stated the long long feedback loops and delivery times to play an important role in initiating the transformation.

4.3 Transformation process

This section describes the transformation processes in each primary study. This answers to the research question 2: *How do large-scale agile transformations proceed?* Table X provides a brief overview of all the transformations.

Table – Transformation process

Facet	**# of organizations**
Transformation initiative	
Top-down	9
Bottom-up	1
Adoption proceeding	
Step-wise	9
Big-bang	1
Framework in use	
Lean-change	2
AM-QuICK framework	1
Scrum of Scrums	1

4.3.1 The initial state

All studied organizations reported their initial software development process to be plan-driven and they referred to the process as waterfall or traditional process. Most studies did not go very deep when describing the old process and focused more on the problems and reasons for the agile transformation rather than the old process description.

Some criticism was present in a few studies. For example, an anonymous product manager stated: "*There are very definite stages, almost monolithic – you have abig design section, big development section, which leads to inefficiencies. Where you find a development problem becomes a design problem that you have to back all the way to those different groups... The requirements capture is proved to be inadequate in traditional methods*". [P5]

4.3.2 The initiation

Most studies reported the transformation to have begun from a management decision. The increased interest in a new and more efficient software development process rose usually from business or process management issues as discovered in section 4.2. Therefore, in most cases the software development team typically had more or less neutral stance on the current process.

In order to promote the transformation efficiently, some companies established dedicated organizations for driving the process. For example, one company created a so-called "Software Process Improvement Group" whose dedicated mission was to align the new agile development process with the new strategic objectives set by senior management [P1]. Another example of this occured at Samsung Electronics, where they established an "Agile Office" to promote agile methods and decrease rigidity organization wide. [P4]

In some cases the transformation was driven heavily by a single person. In two companies a newly hired manager had previously worked with agile methodologies and started to push heavily towards it in the new environment, resulting in initiating the agile transformation [P1, P5].

In one organization the transformation actually begun from the bottom of the organization. According to Ayed at al, their desire to adopt an agile approach came from a single development team in the whole software development organization. The team was seeking to improve their own work and they introduced Scrum in their own team. Subsequently the whole organization adopted Scrum, as it had resulted in increased productivity in the single team. [P2].

As with the aforementioned case, most of the other transformations proceeded in a step-wise manner. Typically, when a team proved the new methodology to be efficient, the natural desire to spread agile values among other teams followed [P2]. In top-down mandated transformations the management typically was a bit cautious at first, and started small with the transformations [P3]. For example, at Samsung Electronics the Agile Office laid down a step-by-step plan to expand the agile culture gradually throughout the organization [P4].

Only one organization started a "big-bang" transformation to agile methods. The new CIO announced

a world-wide transition to begin in the company to move away from traditional software development methods. However, there was still some preliminary research and planning in place even if the transition itself happened in a rapid fashion. As the organization was large and operating on 5 continents, even a "big-bang" transition took over a year to complete. [P5]

4.3.3 Change leaders

Introducing a whole new software development process to an organization requires someone to lead the transformation. Merely announcing a new strategy on its own will not suffice. As such, further investments, training and change leading is required. Each of the primary studies presented their own approaches to leading the agile transformation.

Anwar et al reported that their company, after establishing the Software Process Improvement Group, hired two full-time process improvement engineers to lead the transformation. In addition, they had hired a new software delivery manager that was also pushing for going agile in a lower level. [P1]

At Samsung Electronics the change leadership was appointed to the Agile Office [P4], whose primary mission was to spread knowledge about agile methodologies and organize cross-functional teams to reduce the rigidity of the organization. Therefore, while the initiative itself came from top management, the change actually took place at a lower level. The agile office was organized trainings for all different roles in a Scrum team and supplied all software teams in the company with coaches. [P4]

4.3.4 Communicating the agile model

During the transformation, all organizations invested in communicating the new process and providing training on the subject. In all organizations most people were new to agile, so the prerequisite for the transformation is to provide necessary training on the new model before actually starting the transformation.

Most organizations organized some sort of training as the very first step to the transformation. For example, Anwar et al had reported finding classroom training as the proper option for their needs. This training covered the core concepts of agile, an overview of Scrum (which is the method the company was adopting), release and sprint planning, retrospectives and user story writing. [P1]

Ayed et al had conducted questionnaires to their personnel to identify the most important training needs for their organization. As the transformation in this company had started from the bottom, the software delivery team had a decent understanding of agile to begin with. However, they still organized two half-day training sessions. [P2]

At Ericsson, the transformation was conducted in a rather slow manner. The company first spent half a year to spread knowledge of Agile software development methods before fully launching the transformation. This was found necessary, as the software development organization at Ericsson was the largest of the six primary studies, and therefore the transformation being the most complex one. In order to successfully carry out the transition, they wanted to solve all major problems beforehand. [P3]

The Agile Office at Samsung Electronics first organized training sessions for agile coaches and Scrum masters, who were then distributed to each software development team. The whole software delivery personnel in that company was simply too large to train simultaneously. The agile coaches then supported the Scrum masters in training their respective teams to follow agile methods in their daily work. [P4]

Roman et al reported that their organization approached agile training from top-down as well. Management, development leads and architects promoted discussion on agile, held debate sessions and gave presentations on related topics to increase knowledge on agile methods: *"Local presentation sessions to all members of a certain office were organized to take place world-wide in the same week in which teams already going through the transition were motivated to report their experience to others as a way to encourage the adoption of agile and to share good practices"*. [P5]

4.3.5 Investments

In addition to providing training, all companies invested in the transformation in other ways as well.

Preliminary research. Anwar at al reported the company to have been cautious in avoiding previous improvement pitfalls, and conducting a research of the applicability of the agile process before starting the transformation [P1]. Their research question was *"Can agile process improve the organization's performance indicators while maintining the process maturity level?"*. In order to determine the answer, they conducted an action research and as a result decided to initiate the transformation. [P1]

Ayed et al also reported studying the organization's readiness to adopt agile methods before the transformation. They conducted an interview based study to all relevant stakeholders from different business units to determine their current development process, motivation and reluctance towards adoopting agile software development process. Using this data, they performed a SWOT risk analysis to prepare for possible pitfalls that might occur during or after the transformation. [P2]

Consultants. Two organizations hired consultants to support the transformation process. As stated stated in section 4.3.3, one company had hired two process improvement engineers for the sole purpose of going through the transformation [P1]. Roman et al reported hiring external consultants as agile coaches to closely support the transformation: *"Local coaches were hired to support each of the IT offices. We looked for experienced professionals who have faced similar issues [as] ours in other large corporations"*. [P5]

4.3.6 Piloting

Four of the six organizations made use of some sort of piloting when adopting agile methods. Ayed et al reported finding suitable pilot project that *"is not too risky but is fairly representative of the organizion reality."* After 6 months successful Scrum pilot the company started the full blown transformation covering all teams and organizations. [P2] Duka reported piloting with different agile methods in different teams to gauge the effectiveness and suitability to their environment. Most teams used scrum, but some utilized Kanban. In the end of the transformation their typical approach to software development was

some sort of a combination of the two. [P3] Most valuable insight gained from the pilot projects was to assess possible side effects of re-organizing the teams and overall changes caused by the transformation. [P5]

4.3.7 Introduced practices

As part of the transformation process, some new practices were introduced to organizations' software development processes. Coaching was among the most common practices to be utilized during the transformation. [P3, P4, P5] However, coaching is most valuable during the transformation when the developers are new to agile methods and principles. Other practices that would carry on being made use of include continuous integration, communities of practice and different knowledge sharing practices.

Continuous integration. According to Fowler [9], *"Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily – leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible"*. This approach was adopted at Ericsson as well to increase better means to support faster feedback with the R&D organization. In practice, they built the automated build and test machinery and started co-operating closely with feature development and system integration teams, resulting in faster feature delivery and more streamlined operations. [P3]

Communities of Practice. Another mechanism to increase communication and knowledge sharing was to implement the so-called communities of practice [P5]. This helped teams assess pressing issues with collagues efficiently and increased motivation among employees. A business analyst at the organization commented about the importance of communities of practice: *"[Equally] important as training people is sharing what we are learning with one another, and here is a good way to do it. I hope they do not cut this practice off"*. [P5]

Knowledge sharing practices. Ericsson introduced a multitude of new knowledge sharing practices in order to increase inter-organizational communication and convert tacit knowledge to a written format. Duka reported that they introduced a practice called "Current best thinking" to emphasize the short term planning and to encourage everyone to question and look for alternatives to current processes. Another practice they introduced was an open question session called "Fast Forward Friday", in which everyone at the organization could ask and get answers from the management on the current state of things. These sessions were found valuable by both the employees and the management. [P3]

4.4 Agile scaling frameworks

4.4.1 Lean-change

4.4.2 AM-QuICK

4.4.3 Scrum of Scrums

5 Discussion

6 References

[1] Beck, K., 1999. Embracing change with extreme programming, Computer, 32(10):70–77

[2] Beck, K., et al. 2001. "The Agile Manifesto". http://agilemanifesto.org/

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

[4] Cohn, M., Ford, D., 2003. Introducing an agile process to an organization Computer. 36 (6), 74–78.

[5] Dikert, K., Paasivaara, M., Lassenius, C., 2016. Challenges and success factors for large–scale agile transformations: A systematic literary review. The journal of Systems and Software. p. 87–108.

[6] Kitchenham, B. A., Guidelines for performing systematic literature reviews in software engineering. Technical report EBSE–2007–01, Keele University Technical Report, 2007

[7] Mohammed, A.M., Abushama, H.M., 2013. Popular agile approaches in software development: Review and analysis. In Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International conference on. IEEE, 2013

[8] Rising, L., Janoff, N.S., 2000. The Scrum software development process for small teams. IEEE Software. p. 26–32

[9] Fowler, M. 2006. "Continuous integration". https://martinfowler.com/articles/continuousIntegration.html

[10] Dyba T., Dingsøyr T., 2008 Empirical studies of agile software development: A systematic review. The journal of Systems and Software. 50 (8-9) p. 833–859.

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

[12] Dikert K., Paasivaara M., Lassenius C., 2015, Adopting agile software development in large organizations: A systematic literature review, unpublished manuscript

7 Primary sources

[P1] Anwar A., Kamel A., Ahmed E. 2016. *Agile Adoption Case Study, Pains, Challenges & Benefits*, ACM International Comference Proceedings Series, Association for Computing Machinery, 28–29–May–2016,

60–65

[P2] Ayed H., Vanderose B., Habra N. 2014. *Supported Approach for Agile Methods Adaption: An Adoption Study*, RCoSE 2014 – Proceedings. p. 36–41

[P3] Duka D. 2013 *Adoption of Agile Methodology in Software Development*, MIPRO 2013 – Proceedings, p. 426–430

[P4] Kim S., Lee H., Kwon Y., YuM., Jo H., 2016 *Our Journey to Becoming Agile – Experiences with Agile Transformation in Samsung Electronics*, APSEC, IEEE Computer Society 2017 – Proceedings, p. 377-380

[P5] Roman G., Marczak S., Dutra A., Prikladnicki R., 2016 *On the Agile Transformation in a Large-Complex Globally Distributed Company: Why Boarding this Journey, Steps Taken and Main Foreseen Concerns*, WBMA 2015 – Proceedings, p. 32–39

[P6] Wells H., Dalcher D., Smyth H. 2015, *The Adoption of Agile Management Practices in a Traditional Project Environment: An IT/IS Case Study*, Hawaii International Conference on System Sciences, 2015, p. 4446–4453