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: 14
Responsible teacher: Miia Jaatinen
Supervisor: Maria Paasivaara
Language: English

Abstract:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

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*. 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 developmet 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" (Beck et al 2001).

According to Mohammed and Abushama (2013), 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 (Rising and Janoff 2000). 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 (Beck 1999). The aim is to enable efficient incremental and iterative development using e̊c̊ontinuous feedback from customer, test driven development and continuous integration. Other important practices include pair programming and constant refactoring (Beck 1999).

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 (Dybå and Dingsøyr, 2008). 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 (Lindvall et al 2004). 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 (Lindvall et al 2004, Cohn & Ford 2003)

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 (Boehm and Turner, 2005).

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. 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 exclusion of 43 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 seond 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 six 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 agie transformations [5]. Thus, five of the seven code families used in the aforementioned study were used in the coding of these studies. A description of these families as well as examples 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 5	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 and

4.1.1 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.

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 VI – 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?*

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

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