

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

Author: Ville Kumpulainen
Title: Adopting agile software development in large organizations
Program: Information Networks
Date: 15.9.2017
Page count: 9
Responsible teacher: Miia Jaatinen
Supervisor: Maria Paasivaara
Language: English

Abstract:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur 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

2	Background	4
2.1	Agile software development	4
2.2	Adopting agile methods in large organizations	5
2.3	Definition of large scale agile	5
3	Research Method	6
3.1	Research questions	6
3.2	Research process	6
3.2.1	Inclusion criteria	7
3.2.2	Identification of primary studies	8
4	Results	9
5	Discussion	9
6	References	9

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 development. 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" (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 which emphasize on concurrent planning, implementing, testing and analyzing (Beck 1999). The aim is to enable efficient incremental and iterative development using eg continuous 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 Dingsøy, 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 departments, 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 (2016), 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 electronic 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, i.e. how and why organizations migrate from traditional processes to agile methods. This excludes studies that e.g. compare 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
IEEEExplore	http://ieeexplore.ieee.org	471
ACM	http://dl.acm.org	153
Scopus	http://www.scopus.com/home.url	844
Web of Knowledge	http://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, introduction*, “roll-ou”, rollout
Only software related articles	(software OR (conference=“agile, xp, icgse, ics”)) AND NOT (title+abs=“manufacturin” OR conference=“agile manufacturin”)

4 Results

5 Discussion

6 References

- Beck, K., 1999. Embracing change with extreme programming, *Computer*, 32(10):70–77
- Beck, K., et al. 2001. “The Agile Manifesto”. <http://agilemanifesto.org/>
- Boehm, B., Turner, R., 2005. Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*. 22(5), p. 30–39
- Cohn, M., Ford, D., 2003. Introducing an agile process to an organization *Computer*. 36 (6), 74–78.
- 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.
- Kitchenham, B. A., Guidelines for performing systematic literature reviews in software engineering. Technical report EBSE–2007–01, Keele University Technical Report, 2007
- 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
- Rising, L., Janoff, N.S., 2000. The Scrum software development process for small teams. *IEEE Software*. p. 26–32