Dikert, Kim; Paasivaara, Maria; Lassenius, Casper

Challenges and success factors for large-scale agile transformations

Published in:
Journal of Systems and Software

DOI:
10.1016/j.jss.2016.06.013

Published: 01/09/2016

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of Systems and Software, 119, 87-108. https://doi.org/10.1016/j.jss.2016.06.013

The Journal of Systems and Software 119 (2016) 87–108

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Challenges and success factors for large-scale agile transformations: A systematic literature review

Kim Dikert[a], Maria Paasivaara[a,b], Casper Lassenius[a,b,*]

[a] School of Science, Department of Computer Science and Engineering, Aalto University, Finland
[b] Massachusetts Institute of Technology, Sloan School of Management, United States

ARTICLE INFO

Article history:
Received 6 October 2015
Revised 5 June 2016
Accepted 6 June 2016
Available online 7 June 2016

Keywords:
Agile software development
Organizational transformation
Large-scale agile
Adopting agile software development
Challenges
Success factors
Systematic literature review

ABSTRACT

Agile methods have become an appealing alternative for companies striving to improve their performance, but the methods were originally designed for small and individual teams. This creates unique challenges when introducing agile at scale, when development teams must synchronize their activities, and there might be a need to interface with other organizational units. In this paper we present a systematic literature review on how agile methods and lean software development has been adopted at scale, focusing on reported challenges and success factors in the transformation. We conducted a systematic literature review of industrial large-scale agile transformations. Our keyword search found 1875 papers. We included 52 publications describing 42 industrial cases presenting the process of taking large-scale agile development into use. Almost 90% of the included papers were experience reports, indicating a lack of sound academic research on the topic. We identified 35 reported challenges grouped into nine categories, and 29 success factors, grouped into eleven categories. The most salient success factor categories were management support, choosing and customizing the agile model, training and coaching, and mindset and alignment.

1. Introduction

Agile methods were originally designed for use in small, single-team projects (Boehm and Turner, 2005). However, their shown and potential benefits have made them attractive also outside this context, particularly both for larger projects and in larger companies. This despite the fact that they are more difficult to implement in larger projects (Dybå and Dingsøyr, 2009). Compared to small projects, which are ideal for agile development, larger ones are characterized by the need for additional coordination. A particular problem in applying agile to larger projects is how to handle inter-team coordination. Large-scale agile involves additional concerns in interfacing with other organizational units, such as human resources, marketing and sales, and product management. In addition, large scale may cause users and other stakeholders to become distant from the development teams. Despite these known problems related to large-scale agile, there is an industry trend towards adopting agile methodologies in-the-large (VersionOne, Inc, 2016; Paasivaara et al., 2013, 2014; Dingsøyr and Moe, 2014).

The State of Agile Survey that Version One has been conducting annually since 2007, has recently asked a few questions related to large scale as well, e.g. on scaling methods used and tips for success with scaling agile. According to the latest survey (VersionOne, Inc, 2016), 62% of the almost 4000 respondents had more than a hundred people in their software organization and 43% of all the respondents worked in development organizations where more than half of the teams were agile. Of course, the sample of this study is limited to a selected subset of companies and countries (of the almost 4000 respondents to the latest survey 65% were from North America and 26% from Europe). However, this indicates that there seems to exist a large number of companies that have taken or are taking agile into use in large-scale settings (VersionOne, Inc, 2016).

While the research literature contains several experience reports and some case studies on large-scale agile adoption, a systematic overview and synthesis of this growing body of research is still missing. Freudenberg and Sharp (2010) asked the industrial practitioners at the XP2010 conference to create a backlog of topics they think should be studied. The practitioners voted "Agile and large projects" as the top burning research question. Moreover, among the top ten items three focused on distributed agile development, which is relevant especially for larger organizations as

* Corresponding author.
E-mail addresses: kim-karol.dikert@aalto.fi (K. Dikert), maria.paasivaara@aalto.fi (M. Paasivaara), casperl@mit.edu, Casper.Lassenius@aalto.fi (C. Lassenius).

http://dx.doi.org/10.1016/j.jss.2016.06.013

they are often geographically distributed. In two recent workshops on large-scale agile development organized in XP2013 and XP2014 conferences, adoption of agile methods was one of the highlighted themes needing more research (Dingsøyr and Moe, 2013, 2014).

While research on agile software development is accumulating and maturing, and has provided a basis for conducting systematic literature reviews (Dybå and Dingsøyr, 2008; Jalali and Wohlin, 2012; Senapathi and Srinivasan, 2013; Kaisti et al., 2013), the area of large-scale agile development has not yet been studied through secondary studies. In this paper we start filling in this gap by presenting a systematic literature review of large-scale agile transformations.

2. Background

2.1. Agile software development

Agile software development is a set of iterative and incremental software engineering methods that are advocated based on an "agile philosophy" captured in the Agile Manifesto (Fowler and Highsmith, 2001). While mostly repackaging and re-branding previously well-known good software development practices, the agile movement can be considered as an alternative to so called traditional software development methods. Traditional methods focus on up-front planning and strict management of change, but agile methods were designed to accept and efficiently manage change (Highsmith and Cockburn, 2001; Cockburn and Highsmith, 2001).

Agile methods have been both criticized and advocated, and research has shown that accommodating change may be a factor in both success and failure (Boehm, 2002). It has been shown that agile methods have improved satisfaction of both customers and developers, but on the other hand there is evidence that agile methods may not be a good fit for large undertakings (Dybå and Dingsøyr, 2009). A proposed solution is that each organization seeks its own balance of agile and plan driven methods (Boehm, 2002).

Two of the most popular agile methods are Extreme Programming (XP) and Scrum (Hamed and Abushama, 2013). Scrum is a method focusing on the project management viewpoint of agile development (Schwaber and Beedle, 2002), prescribing timeboxing, continuous tracking of project progress, and customer centricity. The XP development method is a collection of practices for enabling efficient incremental development (Beck, 1999). In practice, many agile development implementations combine the two in some way (Fitzgerald et al., 2006).

2.2. Adopting large-scale agile

The difficulty of introducing agile methods increases with the organization size (Dybå and Dingsøyr, 2008). The difficulty is partly related to size bringing higher organizational inertia which slows down organizational change (Livermore, 2008). Agile development is not founded on the use of individual tools or practices, but rather on a holistic way of thinking. Adopting agile often requires change of the entire organizational culture (Misra et al., 2010).

One significant difference between small and large-scale adoptions is that larger organizations have more dependencies between projects and teams. This increases the need for formal documentation and thus reduces agility (Lindvall et al., 2004). In addition to inter-team coordination, development teams must interact with other organizational units, which are often non-agile in nature. For instance, human resources unit may demand individuals to have strictly specified roles in projects (Boehm and Turner, 2005), or a change control board may inhibit the use of continuous integration or refactoring (Lindvall et al., 2004). All units affected by the agile transformation need to be informed and consulted, and the agile process must be adjusted according to their needs (Lindvall et al., 2004; Cohn and Ford, 2003; Boehm and Turner, 2005).

Agile methods also affect management and business related functions. A key challenge is that management must move away from life-cycle models and towards iterative and feature centric models (Nerur et al., 2005), which requires a change of mindset. The focus must be shifted from long-term planning to shorter term project planning (Misra et al., 2010), as agile methods emphasize that planning is only meaningful for the near future (Cohn and Ford, 2003). However, the lack of planning can be a concern as business and customer relationships often build on long term roadmapping. Enabling operation with shorter term planning requires educating stakeholders and reviewing contracting practices (Boehm and Turner, 2005).

2.3. Definition of large-scale agile

A brief literature search (Dingsøyr et al., 2014) identified previous interpretations of what large-scale agile is. Size had been regarded in terms of size in persons or teams, project budget, code base size, and project duration. The examples of cases that were called "large-scale" included 40 people and 7 teams (Paasivaara et al., 2008), project cost of over 10 million GBP and a team size of over 50 people (Berger and Beynon-Davies, 2009), a code base size of over 5 million lines of code (Petersen and Wohlin, 2010), and a project time of 2 years with a project scope of 60–80 features (Bjarnason et al., 2011). Based on their findings Dingsøyr et al. (2014) ended up measuring large-scale by the number of collaborating and coordinating teams: they categorized as large-scale 2-9 collaborating teams and as very large-scale over ten collaborating teams.

We identified a number of additional studies discussing large-scale agile software development and their interpretations of large-scale. All of these referred to the number of people involved. In early work on agile, Fowler considers the Crystal methodology to be suitable for up to 50 people (Fowler, 2000). The same number has been reported as seen by practitioners and researchers as the size of the largest organization suitable for agile (Williams and Cockburn, 2003). Other studies have referred to agile projects including up to 50 people as small (Koehnemann and Coats, 2009), and considered a development project large if it had a staff between 50 and 100 people, including all project personnel (Elshamy and Elssamadisy, 2006). The largest numbers were 300 people across 3 sites (Moore and Spens, 2008). Participants of the XP2014 large-scale agile workshop gave very varying definitions for large-scale agile development (Dingsøyr and Moe, 2014), showing that what is seen as large-scale depends very much on the context and the person defining it.

Based on these findings, we defined large-scale to denote *software development organizations with 50 or more people or at least six teams*. All people do not need to be developers, but must belong to the same software development organization developing a common product or project, and thus have a need to collaborate. For instance, Scrum masters and software architects are counted when assessing the organizational size. As some studies present the number of teams rather than the number of people, we correspondingly defined large-scale to denote development efforts involving at least six teams. Having six teams with an average size of six to seven people, plus a number of supporting staff, can reasonably be considered to form an organization of 50 people. In this definition, we include both companies that as a whole focus on software development, as well as the parts of larger (non-software focused) organizations that develop software, e.g. in-house software development units of large non-software corporations.

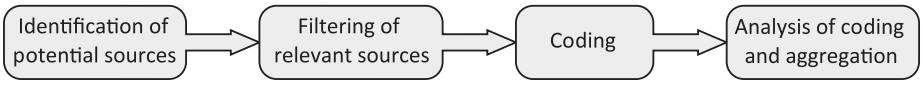

Fig. 1. Outline of the research process.

Table 1
Inclusion criteria.

Facet	Relevant topics	Examples of non-relevant topics
Agile software development	The organization develops software; the introduced method is agile	Agile manufacturing; Scrum in management boards
Organizational transformation	Presents insights about the transformation process	Comparison of before and after; how agile is currently used in large scale
Large scale	Agile practiced by a development organization of at least 50 people or 6 teams	Scaling up from small; a single agile team in a large setting
Empirical	Case studies, experience reports	Textbooks, student experiments, theory papers

3. Research method

3.1. Research questions

Systematic literature reviews are a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest. They are appropriate for summarizing existing research, for identifying gaps in the existing literature, as well as for providing background for positioning new research (Kitchenham, 2007).

In this paper, we present the results of a systematic literature review on the topic of large-scale agile transformations. The review is positioned in, and utilizes the literature of, the field of software engineering. In the review, we study the following research questions:

- RQ1: What challenges have been reported for large-scale agile transformations?
- RQ2: What success factors have been reported for large-scale agile transformations?

While large-scale agile transformations could provide many viewpoints to use as research questions, we chose these two questions because we believe they represent the viewpoints which are likely to provide actionable insights to practitioners as well as researchers. The research questions are not intended to present complements to each other, but rather two distinct viewpoints that can highlight different characteristics in the transformations.

3.2. Research process

The research process consisted of four main stages depicted in Fig. 1. The selection of primary studies was done in two stages, first using keyword-based database searches to identify potentially relevant sources, and then manually filtering the search result. The manual filtering process was executed independently by the two first authors. Data extraction was done by qualitative coding of the selected primary studies by the first author. Finally, the results were elicited by aggregating and analyzing the coding of the primary documents. The entire process was audited and mentored by the third author.

3.2.1. Inclusion criteria

Based on the research questions and focus of the research, we defined four facets to guide our inclusion/exclusion decisions: *agile software development, organizational transformation, large-scale*, and *empirical*. Table 1 lists the facets and gives examples on matching topics and non-relevant topics. For a study to be included it needed to be relevant on all facets.

Examples of topics excluded by the facet of *agile software development* are agile manufacturing and the application of Scrum outside software engineering, e.g., in management boards. In addition, we required that the organizational transformation was aimed at introducing agile methods, which excluded other development methods (Sagesser et al., 2013), and the use of agile methods in other contexts than software development (Hodgkins and Hohmann, 2007).

The facet of *organizational transformation* was interpreted so that the primary study had to present insights on the process of an agile transformation. Examples of excluded topics include comparing the original and agile development methods (Petersen and Wohlin, 2010), discussing the use of agile in a large enough organization but not describing how the new methods were introduced (Mishra and Mishra, 2011), and merely presenting agile tools in large-scale use (Kim and Ryoo, 2012).

Transformation and "scaling up" of agile practices in use are very closely related concepts, and in some cases they are one and the same. For instance, if transformation begins with a pilot and then is gradually rolled out in the organization, the process could well be seen as a "scaling up" journey. These kinds of journeys are included in our study. However, we excluded cases where an initially small agile organization grew organically (Maranzato et al., 2012), and discussions focusing on processes or tools without describing organizational change (Lyon and Evans, 2008).

The facet of *large scale* was interpreted as discussed in Section 2.3: a single software development organization with at least 50 persons or six teams. In some cases, the source presented very vague indicators on size. For instance, one case (Cloke, 2007) was included, as there were indications of large-scale considerations, although the size remained unclear. If there were no indications of size, the paper was excluded, e.g. (Miller and Haddad, 2012).

To complicate matters, some papers talked about "the team" in singular when referring to the organization (Hodgkins and Hohmann, 2007), making it nontrivial to judge whether the organization met the large scale criteria based on the choice of words of the author.

Further examples on exclusion by the *large scale* facet were cases with large organizations but only a single team adopting agile (Fulgham et al., 2011). We considered cases of single teams (although as part of a larger organization) unrelated to this research as our focus is on transformation of the entire (development) organization.

Also piloting cases that reported only single teams using agile, even though considering the whole organization would meet the large scale criteria, were excluded, e.g. (Scott et al., 2008). Finally, cases where the organization was growing to large scale, but did not meet the size criteria at the start of the transformation, were excluded.

According to the facet of *empirical* we excluded studies that did not discuss a distinguishable real world case. We excluded

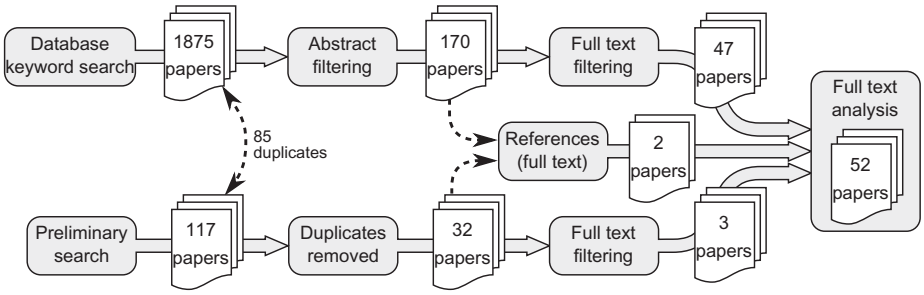

Fig. 2. The study selection process.

Table 2
Databases included in search, and number of matched articles.

Database	URL	# of matches
IEEExplore	http://ieeexplore.ieee.org	745
ACM	http://dl.acm.org	168
Scopus	http://www.scopus.com/home.url	1596
Web of knowledge	http://apps.webofknowledge.com	786

textbooks, studies merely presenting theories, as well as studies that did not include any case organization. Moreover, we excluded studies on the benefits or limitations of agile in general. Lastly, we excluded student experiments as it is implausible to draw conclusions on the organizational dynamics in our large-scale context based on experiments.

3.2.2. Preliminary searches

Before proceeding with identifying the primary studies a number of preliminary searches were performed. The purpose of the preliminary searches was to develop and evaluate different search strings. In addition, we used the searches to identify a set of relevant papers that should be matched by the actual search. We started by examining top ranked hits by trivial keywords that the more complex final search string might miss. Initial searches were made using keywords that were as general as possible, including "agile transformation" and "large scale agile". Based upon these preliminary searches, we selected 117 papers that seemed relevant based upon the title. We used this set as a "sanity check" when developing the actual database search.

3.2.3. Identification of primary studies

The gathering of potential primary studies was based on a search in the online databases listed in Table 2. We constructed a search string based on the previously discussed facets. However, our preliminary searches showed that picking keywords with good precision was difficult. In particular, we did not succeed in representing the facets *large scale* and *empirical* with precise words. Therefore, we included only the facets *agile software development* and *organizational transformation* in the keyword search, with the consequence of increasing the manual filtering effort in the subsequent steps. The used keywords are shown in Table 3.

3.2.4. Study selection

The set of potential primary studies identified by the database search was refined in two steps, first by filtering based on abstracts and finally based on the full text. The study selection process is outlined in Fig. 2.

The database keyword search matched 1875 unique papers. The abstracts of these papers were categorized independently by the two first authors into three categories: include, exclude, and uncertain. There were 1578 exclusions and 62 inclusions that both researchers agreed on. The inclusion decisions for the 235 abstracts with disagreement or uncertainty were resolved through discussion. At this stage papers were excluded only if both researchers deemed it clearly irrelevant, including any uncertain cases for full text filtering. As a result 170 papers were selected for full text filtering.

Full text filtering was performed by evaluating the text of each article against the four facets of the inclusion criteria. Filtering was done in two steps. In the first step, the first author extracted data relevant to the four facets. Based on the extracted data, 76 papers could be immediately deemed as included or excluded. The remaining 94 papers were evaluated against the inclusion criteria by the two first authors, and a decision was made after discussing each paper separately. In difficult cases, the third author was consulted to reach a decision. As a result of the full text filtering, 47 papers were selected for inclusion.

We evaluated the result of the database keyword search against the benchmark created in the preliminary search step. We concluded that 85 of the 117 preliminarily selected papers were matched by the database keyword search. The missed 32 preliminary papers were examined, resulting in including three additional papers as primary sources.

In parallel with the full text filtering step the references of all 170 papers selected for full-text filtering were also examined for relevance. Most papers used references very scarcely, typically referencing well known descriptions of agile methods. This step led us to include two additional papers in our full text analysis.

Table 3
Facets and related search terms used.

Facet	Keywords
Agile methods (before & after)	`agile, scrum, ''extreme programming'', waterfall, ''plan-driven'', RUP`
Organizational transformation	`transform*, transiti*, migrat*, journey, adopt*, deploy, introduc*, ''roll-out'', rollout`
Only software related articles	`(software OR (conference=''agile, xp, icgse, icse'')) AND NOT (title+abs=''manufacturing'' OR conference=''agile manufacturing'')`

Table 4
Context information extracted from primary studies.

Contextual code	Explanation
Agile method	Agile methods used in the organization (e.g. Scrum, XP)
Business area	The business area in which the organization operates.
Organization size	Mentioning the size of the case organization.
Time of transformation	When has the transformation been in progress, how long has the transformation taken.
Research process	The paper describes a research process.
Geographical location	Where the organization is located.
Large scale definition	A definition of large-scale software development.
Multisite / GSD	Mentioning of a multisite organization.

Table 5
Code families, codes and quotations.

Code family	Description	Examples	Codes	Quotations
Reason to change	Reasons to start the transformation	Demand for faster delivery	30	123
Transformation process	Statements describing the transformation process	Top-down, big bang, step-wise	16	580
Challenges	Statements presenting challenges in the transformation	Change resistance	40	323
Success factors	Statements presenting success factors in the transformation	Management support	44	260
Investing in change	Factors presenting how the organization is investing in the transformation	Training, consultants, tools	5	137
Practices	Practices used or estab- lished during the trans- formation, but described as neutral wrt. to successes and challenges	Coaching, pilot- ing, continuous integration, communities of practice	11	170
Contextual	Contextual codes defined in Table 4	Agile method, organization size, large-scale definition	8	215
		Total	154	1575

As a result, we selected a total of 52 papers for inclusion in the analysis stage.

3.2.5. Handling of duplicate reports on a single case

In several cases, more than one primary study discussed the transformation of the same organization. Duplicate descriptions typically focused on different aspects. For example, one paper would highlight the viewpoint of the developers (Fry and Greene, 2007), and another would consider the transformation from the user experience designers' point of view (Federoff and Courage, 2009).

Even if the transformation of a single organization was described in many studies, all sources that passed the inclusion criteria were included. Studies presenting the same organization were treated as one unit so that we could gain as much insight on each organization as possible. Conversely, there were also a few papers that presented multiple case studies, and in those cases we treated each studied organization individually.

Instead of using the most complete paper as suggested in the guidelines for systematic literature reviews (Kitchenham, 2007), we combined the results presented in each paper and considered the case as a single unit in our analysis. Keeping in mind the potential bias caused by duplicate publications, we think that including all papers enabled us to get a more in-depth understanding of the individual cases.

As a result, we identified 42 unique organizations in the primary studies. We use the term *study* to refer to the primary publications, and the term *case* to refer to an individual case organization that may be described in several different studies.

3.2.6. Study quality assessment

The primary studies for this literature review are almost exclusively industry experience reports. We identified only six case studies with a clearly defined research method, and observations on transformation were presented only as a minor part in these studies. Based on this finding we conclude that case studies presenting insights into large-scale agile organizational transformations are very scarce. We deemed that the results would be distorted heavily and many valuable studies would be left out if a strict quality assessment would be part of the inclusion criteria. As a result we decided to include all experience reports, regardless of the potential problems of author and publication bias.

3.2.7. Coding of primary studies

We coded the primary studies using an integrated deductive and inductive approach, coding both a contextual and a findings part (Cruzes and Dyba, 2011). We established an a priori list of codes for contextual information (Table 4), which included the codes agile method used, business area, organization size, time of transformation, research process used, geographical location, definition of large-scale used in the paper, and multisite case. Codes related to the research questions were created by an inductive process to avoid having our previous assumptions affect the choice of codes.

To guide inductive coding, we established five different code families a priori: reasons to change, transformation process, challenges, success factors, and contextual. The four first families were related to our overall research questions. In addition to these five evident families two additional families were established during the coding process: investing in change, and practices. A description defining the scope and containing/examples of codes were defined for each family. The example codes would not necessarily be used as actual codes, and would not represent the entire or final scope of the families. Table 5 presents the codes families in the final state of the coding. Table 5 also shows the total number of codes and quotations created in the coding process. The total number of quotations is less than the sum of quotations in the categories because a single quotation may have multiple codes. Note that in this paper, we only discuss the results related to challenges and success factors, and that it as such forms a part of a larger study.

3.2.8. Synthesis of primary studies based on coding

We synthesized our findings by creating an initial organization of codes into high level categories based on the code labels. Each code was classified into a single category.

After assigning each code to a unique category, the content of the categories was studied. Each category was studied by reading through each quotation of each code included. Typically the quotations were displayed and examined in their original context, considering the surrounding paragraphs of the quotation. Notes

Table 6
Author affiliations.

	Affiliation type	Papers
Experience reports	Internal to organization	32
	Consultant and internal to organization	7
	Consultant	5
	Researcher	2
Case studies	Researcher	4
	Researcher and internal to organization	1
	Internal to organization	1

were made on each quotation presenting noteworthy observations. We used the notes to create a concise description for each code.

We then refined the categories based upon the concise descriptions. Some codes were re-categorized, and the definitions of a few categories were revisited, until a final ordering was reached. The results are presented according to the final categorization.

The tables presenting the categories of challenges and success factors list references to the primary studies and case organizations. We have included a count of how many individual cases mentioned each factor, and a percentage of this count relative to the 42 individual cases. The purpose of these counts is to give the reader a sense of how often the factor was mentioned. We want to note that merely the frequency of mentioning does not make as generalizable evidence for the actual importance of the factors.

4. Results

In this section we present our findings. First, we present an overview of the primary studies. Then, we discuss the findings organized according to our research questions.

4.1. Overview of the primary studies

In this section we present the type of the primary studies, some characteristics of the case organizations, as well as the agile methods used by the cases.

4.1.1. Study types

The results include findings from 52 primary studies discussing how 42 different software organizations introduced large-scale agile methods. Most studies were experience reports (46 studies). Only six of the included studies had a research method explicitly stated. The publication forums of the primary studies were distributed so that 47 sources were conference proceedings, four sources were journal articles, and one source was a technical report.

In most cases it was evident that the author had been personally involved in the transformation. In 41 out of the 52 papers one or more authors had a direct affiliation to the case organization. Table 6 summarizes the affiliations of the authors. Internal affiliation refers to cases where the author is employed by the case organization. In some cases there are multiple authors with different affiliations.

All included studies and transformations were dated after year 2000. There were peaks in transformation studies in 2008 and 2011. The studies were typically published two years after the start of the transformation.

Tables 7 and 8 list the case organizations and primary sources. Some key contextual information on the organizations is included in the tables. We reference the primary studies with a capital P and a number, and the unique case organizations with a capital C and a number.

Table 9 summarizes the content of included research papers. Whereas none of the studies had their research focus directly on the transformation process, they provided similar information as shallow experience reports to our study.

In the above mentioned tables we have evaluated the relevance of each primary study with respect to our research questions with a subjective score. We defined the scores as follows. 1: some sentences revealing factors relating to transformation. 2: a couple of paragraphs describing transformation. 3: Several paragraphs throughout the paper explaining transformation. 4: entire sections giving a narrative of the transformation. 5: the entire paper focuses on describing how the transformation proceeded.

4.1.2. Case organizations

The size of the organizations varied from the minimum included size of 50 to 18,000 people. The median size was 300 people. In seven studies the size was presented in terms of teams, ranging from the minimum of six teams to 150 teams. The median was 10 teams. In eight studies there was no direct indication about the size but the issues discussed revealed that the organization was of a large size according to our definition.

Different business areas were represented quite evenly. Online services was the largest group, including companies providing software as a service solutions for businesses, online media players for consumers, online services for consumers, and communication software for businesses. The second largest group was telecommunications, including companies such as British Telecom, Cisco, Ericsson, and Nokia Siemens Networks. The third largest group was enterprise management solution providers with products for business process management, project portfolio management, and facility management. Table 10 summarizes the business areas and maps them to cases.

4.1.3. Agile methods used

The most prevalent agile method used in the transformed organizations was Scrum, which was the sole agile method mentioned in 25 cases. The second most mentioned agile method was Extreme Programming (in 4 cases), which was often combined with Scrum (in 5 cases). Lean software development was mentioned in 6 studies, although in all cases in combination with Scrum. Other agile methods mentioned were Unified Process[1], Adaptive Development Methodology (ADM), and Rapid Application Development. In one case the agile method was not named.

It was quite common that organizations sought to combine agile methods. Especially Scrum, XP, and Lean software development were used together. In addition, many cases mentioned use of XP practices, such as test driven development and continuous integration, without explicitly stating XP as the process being used. Combining and customizing agile practices was also evident as many organizations viewed the agile method as continuously evolving. Organizations evolved the agile methods for instance through retrospectives and continuous improvement.

4.2. Transformation challenges

Any organizational transformation that involves numerous individuals will face challenges. In this section we answer our first research question, RQ1: *What challenges have been reported for large-scale agile transformations?*

We organized the found 35 challenges, each mentioned by several sources, into nine categories. The categories are summarized in Table 11.

4.2.1. Change resistance

General resistance to change. People are not willing to change unless there are good reasons that they understand, and the

[1] We acknowledge that it can be debated whether Unified Process is an agile method. However, since the author listed it as such, we list it here.

Table 7
Experience reports

Case	Paper(s)	Company	Business area	SW org size	Initial state	Agile methods	Trans-from start	Relevance
C02	P03	Amazon.com			Small autonomic teams	Scrum	2004	5
C03	P04	BEKK consulting	IT- and management consulting	160	Waterfall	Unified Process	2003	3
C04	P05, P17	BMC Software	Distributed Systems Management business unit	300	Waterfall, traditional, dispersed	Scrum	2004	5, 5
C05	P06, P09	Yahoo!	Online services	150 teams	Waterfall (named PDP), gated, city-states	Scrum, Lean	2005	4, 5
C06	P07, P52	Nokia Siemens Networks		500		RAD, Scrum	2007	3, 2
C07	P08	ABC Bank	Banking	300		Scrum, UP, XP. Custom delivery		3
C08	P10	Yahoo! Music	Media player for consumer		Enforced top-down, heavy-weight, BDUF, waterfall	Scrum	2006	4
C09	P11	Unisys Cloud Engineering			Waterfall	Scrum	2010	5
C10	P12	BBC, iPlayer	Media	10 teams	Independent processes, Heavy practices	Scrum		4
C11	P13, P15, P16, P28, P33	Salesforce.com	Online services, SaaS CRM software	30 teams	Loose waterfall-based process	Scrum, XP, Lean, ADM	2006	2, 2, 5, 2, 2
C12	P14	MyBoeingFleet.com	Extranet website		CMMI, Macroscope, gate-based	Scrum	2007	3
C13	P18	Ericsson Netherlands	Telecommunications	300		Scrum, XP, Lean	2007	4
C14	P19	Citrix Online	Conferencing and screen sharing software	44 teams	RUP waterfall	Scrum	2007	1
C15	P20, P21	Anonymous	Government	50	Waterfall	Scrum	2010	3, 3
C16	P22	Ericsson R&D Finland	Telecommunications	400	Traditional, functional, silo-based	Scrum, Kanban, Lean		2
C17	P23, P32	British Telecom	Telecommunications	14000	Old fashioned process-driven; Waterfall	Mentioning of Scrum, XP	2004	3, 2
C18	P24	SoftwarePeople			CMMI level 3	Scrum	2006	4
C20	P26	Nokia	Telecommunications		Waterfall, some Scrum on team level	Scrum	2007	2
C21	P27	Anonymous	Healthcare	300	Waterfall	Scrum, XP	2005	3
C22	P29	Microsoft IT	IT services	9 teams	Waterfall and PMI methodologies	Scrum	2006	3
C23	P30	Healthwise	Healthcare, DSS system	200	Expanding organization	Scrum, Lean	2004	3
C24	P31	Borland		300		Scrum	2006	4
C25	P34	Siemens Medical Solutions USA	Health services	300		Scrum		4
C27	P36	Anonymous	Financial services	100	Traditional, phase-based approach	XP	2001	4
C28	P37, P38	Gale	Online services	6 teams	Waterfall, overall unclear		2009	3, 3
C29	P39 (T1)	Anonymous	Online services	50	Waterfall	Scrum		3
C30	P39 (T2)	Anonymous	Banking	9 teams		Scrum		3
C31	P40	Quintiles Trans- national Corp			Homegrown waterfall, CMMI, silos	Scrum		3
C32	P41	Nyx		270	Traditional waterfall-style (RUP)	Scrum	2007	4
C33	P43	VeriSign Enterprise Security	Information security	50		Scrum	2006	4
C34	P44	Primavera Systems	Enterprise project portfolio management	90	Waterfall	Scrum	2003	3
C35	P45	SAP AG	Enterprise applications	18000	Waterfall-like process	Scrum, Lean	2006	4
C36	P46	KeyCorp	Financial	1500	Waterfall, command-and-control	Scrum	2005	5
C37	P47	Capital One				Scrum	2004	3
C38	P48	Cisco Voice Technology Group	Telecommunications	1500	Waterfall	Scrum		4
C39	P49	Qwest Communications	Telecommunications	5000	Rigorous, collaborative in one unit, CMM	XP	2002	5
C40	P50	Pegasystems	Business process management	200	Traditional project management	Scrum	2009	3

Table 8
Case studies.

Case	Paper(s)	Company	Business area	SW org size	Initial state	Agile methods	Transfrom start	Relevance
C01	P01, P02	Anonymous, HQ in the UK	Telecommunications	7500		XP	2005	3
C16	P42	Ericsson R&D Finland	Telecommunications	400	Traditional, silo-based	Lean, Scrum	2009	2
C19	P25	Nokia Siemens Networks	Telecommunications	150	Plan-driven, waterfall	Practices (XP), Scrum	2010	1
C26	P35	Anonymous		50	Waterfall	XP		2
C41	P51 (C1)	Anonymous	Messaging services	100		Scrum	2008	2
C42	P51 (C2)	Anonymous	Facility management	300	Prince2, Waterfall	Scrum	2004	2

Table 9
Summaries of case study findings.

Case	Study focus	Key results the study focuses on	Subjective relevance	Source material
C01	The effects of agile transformation.	Elements of interpretation and practice.	Focus not on the transformation process.	7 interviews, observed 4 meetings and two days an agile team, documents
C16	How Lean thinking is implemented in software development companies	The current state of practice reflected against a Lean framework	The focus mostly on the current state instead of the transformation process.	17 focus group sessions, material walkthrough workshop, process documents. Sessions involved 21 company representatives.
C19	Has visibility, reaction speed, quality, and motivation changed; comparing before and after	Changes in indicators during the transformation	Focus not on the transformation process.	Defect data, survey responses, comments from feedback sessions after the surveys
C26	Good and bad aspects of communication, during an agile transformation	Brief narratives from a multitude of perspectives evaluating the transformation	Relevant but very brief narratives providing some insights on the transformation process.	Survey questionnaire including open ended questions. Further clarifications via Skype or email.
C41, C42	Multiple-case study on analyzing Scrum introduction paths.	Brief narrative through the steps preparation, introduction, and customization.	Quite relevant, but very brief one paragraph narrative.	3-5 interviews per case.

Table 10
Business areas.

Business area	# of cases	Case reference	Primary source reference
Online services for consumers and business	9	C5, C8, C10, C11, C12, C14, C28, C29, C41	P6, P9, P10, P12, P13, P15, P16, P28, P33, P14, P19, P37, P38, P39, P51
Telecommunications	8	C1, C13, C16, C17, C19, C20, C38, C39	P1, P2, P18, P22, P23, P26, P32, P25, P48, P49
Enterprise management	5	C3, C34, C35, C40, C42	P4, P44, P45, P50, P51
Banking	4	C7, C27, C30, C36	P8, P36, P39, P46
Healthcare	3	C21, C23, C25	P27, P30, P34
IT services	2	C4, C22	P5, P17, P29
Government	1	C15	P20, P21
Information security	1	C33	P43
Not specified	9	C2, C6, C9, C18, C24, C26, C31, C32, C37	P3, P7, P52, P11, P24, P31, P35, P40, P41, P47

change is perceived easy enough. It is difficult to attain a buy-in for a change, and even organizations that have a flexible culture will face resistance [P10]. It should be expected that not everyone will be willing to change, even so that some employees will never adapt to the new way of working [P6]. During a transformation period, objections to change may become a major reason for loss in time and productivity [P29].

Numerous reasons for change resistance were reported. For instance, Fecarotta [P14] described the organization of Boeing as risk-averse and cautious. Any change was further hindered by a deeply rooted status quo and high employee retention [P14]. In some cases it was reported that people worried about new roles and responsibilities that agile might bring [P44]. For instance, testers were worried that they would need to take on cross-functional tasks, which would be outside their area of expertise [P18]. Yet another reason for change resistance was the move from individual offices to team areas [P22]. People felt that they were being monitored more because of the increased level of interaction within the team and between the various project stakeholders [P45].

Skepticism towards the new way of working. Skepticism and distrust in agile development in general were other common problems. Seffernick describes how management did on the one hand acknowledge the benefits of agility, but on the other hand opposed its introduction due to contract reasons, the current matrix organization, and other organizational practices [P46]. Lewis reports that tending to skeptical team members wasted time and effort [P29]. Skepticism was often created by misconceptions, including that agile does not work for complex products [P5, P36], agile needs to be implemented in a prescriptive by-the-book way [P9], that frequent meetings will cause overhead [P18], and that agile equals avoiding governance and working without a plan [P8].

With skepticism we refer to any kind of reservedness towards the new way of working. While reservedness might not be a problem on its own, it is still worth to acknowledge that reservedness is mentioned in many cases.

Top down mandate creates resistance. The way the transformation is initiated affects how change resistance will show. In several cases, the change initiative came from management, and when presented in a bad way, became a mandate that people were not receptive for. Spayd [P49] summarizes this as: *"Organizations do not change merely because the boss says so, at least not in the way that is intended"*. In that organization the introduction of CMM had been carried out consistently by a mandate, but the collaborative nature of agile did not mix well with such a mindset. A top-down mandate may also dilute the understanding of the reasons for starting the transformation and the understanding of agile development

Table 11
Challenges.

Challenge type	Primary sources	Case organizations	# of cases
Change resistance 16 (38%)			
General resistance to change	P14, P18, P22, P44, P45	C12, C13, C16, C34, C35	5 (12%)
Skepticism towards the new way of working	P5, P8, P9, P18, P29, P36, P46	C4, C5, C7, C13, C22, C27, S36	7 (17%)
Top down mandate creates resistance	P2, P12, P37, P49	C1, C10, C28, C39	4 (10%)
Management unwilling to change	P2, P3, P6, P49	C1, C2, C5, C39	4 (10%)
Lack of investment 13 (31%)			
Lack of coaching	P1, P6, P23, P45, P47, P49	C1, C5, C17, C35, C37, C39	6 (14%)
Lack of training	P11, P20, P45	C9, C15, C35	3 (7%)
Too high workload	P37, P42, P49	C16, C28, C39	3 (7%)
Old commitments kept	P11, P21	C9, C15	2 (5%)
Challenges in rearranging physical spaces	P6, P14, P29, P36, P38, P49	C5, C11, C22, C27, C28, C39	6 (14%)
Agile difficult to implement 20 (48%)			
Misunderstanding agile concepts	P6, P26, P37, P38, P42, P44, P45, P48, P49	C5, C16, C20, C28, C34, C35, C38, C39	8 (19%)
Lack of guidance from literature	P5, P6, P10, P13, P21, P27, P30, P45, P49	C4, C5, C8, C11, C15, C21, C23, C35, C39	9 (21%)
Agile customized poorly	P9, P29, P44, P46	C5, C22, C34, C36	4 (10%)
Reverting to the old way of working	P6, P11, P12, P18, P21, P38, P44, P49	C5, C9, C10, C13, C15, C28, C34, C39	8 (19%)
Excessive enthusiasm	P4, P6, P12, P20	C3, C5, C10, C15	4 (10%)
Coordination challenges in multi-team environment 13 (31%)			
Interfacing between teams difficult	P9, P10, P13, P17, P24, P26	C4, C5, C8, C11, C18, C20	6 (14%)
Autonomous team model challenging	P7, P13, P29, P51	C6, C11, C22, C41	4 (10%)
Global distribution challenges	P29, P45, P48	C22, C35, C38	3 (7%)
Achieving technical consistency	P5, P24, P26, P27, P50	C4, C18, C20, C21, C40	5 (12%)
Different approaches emerge in a multi-team environment 9 (21%)			
Interpretation of agile differs between teams	P8, P9, P38, P43, P48	C5, C7, C28, C33, C38	5 (12%)
Using old and new approaches side by side	P1, P8, P10, P26, P29	C1, C7, C8, C20, C22	5 (12%)
Hierarchical management and organizational boundaries 14 (33%)			
Middle managers' role in agile unclear	P2, P11, P27, P30, P38, P49, P52	C1, C6, C9, C21, C23, C28, C39	7 (17%)
Management in waterfall mode	P3, P6, P35, P38, P45, P46	C2, C6, C26, C28, C35, C36	6 (14%)
Keeping the old bureaucracy	P20, P48	C15, C38	2 (5%)
Internal silos kept	P10, P11	C8, C9	2 (5%)
Requirements engineering challenges 16 (38%)			
High-level requirements management largely missing in agile	P24, P31, P39, P45, P51	C18, C24, C29, C35, C41	5 (12%)
Requirement refinement challenging	P1, P17, P33, P48	C1, C4, C11, C38	4 (10%)
Creating and estimating user stories hard	P5, P11, P27, P30, P33, P37	C4, C9, C11, C21, C23, C28	6 (14%)
Gap between long and short term planning	P9, P10, P13, P26, P31, P38	C5, C8, C11, C20, C24, C28	6 (14%)
Quality assurance challenges 6 (14%)			
Accommodating non-functional testing	P17, P28, P31, P48	C4, C11, C24, C38	4 (10%)
Lack of automated testing	P10, P11, P17	C4, C8, C9	3 (7%)
Requirements ambiguity affects QA	P5, P48	C4, C38	2 (5%)
Integrating non-development functions 18 (43%)			
Other functions unwilling to change	P1, P2, P5, P9, P10, P17, P18, P28, P31, P38, P42, P45, P46, P48, P50	C1, C4, C5, C8, C11, C13, C16, C24, C28, C35, C36, C48, C40	13 (31%)
Challenges in adjusting to incremental delivery pace	P6, P9, P10, P15, P17, P28, P46, P48, P50	C4, C5, C8, C11, C36, C38, C40	7 (17%)
Challenges in adjusting product launch activities	P31, P38	C24, C28	2 (5%)
Rewarding model not teamwork centric	P3, P6, P38, P40, P42, P49	C2, C6, C16, C28, C31, C39	6 (14%)

overall [P2]. O'Connor describes how team members became skeptical towards agile when implementing the mandated changes did not bring any visible benefits [P37]. A problem relating to a top-down mandate is that if management does not define a clear goal for using agile, developers may feel that the agile methods may be replaced by something else at any time [P12].

Management unwilling to change. Change resistance amongst managers can also create problems. In cases where the change was emerging bottom-up, management became reluctant to change, making significant organizational change above the team level impossible. For instance, Atlas [P3] describes how executive support would have been required to extend the agile process to involve the product management office and separate quality assurance groups. In this case, when managers were not involved in the transformation, the agile way of working could not spread beyond the development teams [P3]. Lack of middle management support for change and a disinclination to change management culture were seen as some of the most serious problems in the transformation [P2, P49]. Middle management is in a position to undermine the entire transformation, and may do so if they do not participate in, and thus understand, the agile method. Agile brings changes to some management roles [P2], and lack of understanding of agile development will leave managers feeling left out [P6].

4.2.2. Lack of investment

Lack of training. Training and coaching are direct investments in transformation and their lack is an evident problem. Not providing enough funding for these activities can create difficulties in the transformation [P11]. Hajjdiab [P20] describes how reluctance of management to invest in training caused teams to be ill prepared for the transformation effort, eventually resulting in ending the use of agile methods. A less dramatic outcome of too little training was lower motivation [P45].

Lack of coaching. Arranging proper coaching was a problem in many organizations. It is critical to coach teams in their real work environment as a proper change in mindset is difficult to achieve only by attending training sessions [P9, P23].

In cases where numerous teams needed to be coached the demand often exceeded the capacity of available coaches [P1, P6, P23, P49]. Lack of coaching was also attributed as a reason why the success of pilot teams could not be repeated when agile was adopted more widely [P9, P45]. Silva and Doss [P47] described that when it was hard to fill the coaching positions, people less seasoned in agile were appointed as coaches, which created the risk that agile practices would not be taught correctly.

Too high workload. Even though the case organizations were in a state of transformation, the workload of the teams was not always adjusted to facilitate the change process. Some organizations started their agile journey in a state where people were overcommitted, and later realized that overloaded people will not be able to change their behavior and learn new ways of working [P37, P42].

Old commitments kept. Sometimes all old commitments were kept despite the transformation. In one case, management forced people to remain committed to firm deadlines, even midst in the organizational transformation [P21]. The engagement in old commitments and tasks resulted in ignoring new agile practices, and eventually the agile method broke down [P21]. One case [P11] describes how time was allocated for tending to old commitments. However, even though they had prepared for extra work, people with specific expertise became overloaded.

Another case describes how senior management pressed teams to deliver what the customer had been promised, regardless of what the current velocity of development predicted. The pressure to deliver interfered with the transformation, but also forced a change in the old way of working [P49].

Rearranging physical spaces. In some cases, the old way of working had people seated physically distributed, in opposition to what agile methods suggest. Office spaces needed to be arranged so that the entire team could work in a single room, but it took time and effort, and it was not always possible [P6, P14, P36, P38]. Lewis and Neher describe how dedicated rooms for teams could not be arranged, and team events such as daily stand-ups required booking a conference room [P29]. In another case, the facilities organization shut down the teams' attempts to modify their working spaces [P49].

4.2.3. Agile difficult to implement

A common challenge was that implementing agile methods turned out to be difficult. An experienced software team may do well in training, but when the time comes to apply agile techniques in practice, the team may get lost [P21].

Misunderstanding agile concepts. There were many examples of problems caused by misconceptions of what agile software development is. On a general level, Bang [P4] describes how the values of the agile manifesto were not understood, and agile practices were carried out without understanding their purpose. In one case, management saw the purpose of agile simply being faster product delivery [P9].

Examples on the team level include how agile was seen as the freedom to hack without documentation [P26], that development could be done without design tasks [P37], and that agile meant that everyone should become a generalist [P42]. Schatz and Abdelshafi describe how teams presented unfinished work at reviews and ignored the principle of showing only completed increments, which resulted in a backlog of bugs [P44]. In one case, team members had perceived the introduction of agile as a means to squeeze out more efficiency [P45]. In addition, the flatter organization was seen as fewer career opportunities, pushing team members to compete for visibility [P45].

A further misunderstanding of agile was due to viewing it only through the tools used, such as management software [P48]. Superficially focusing only on the tools themselves and not the reasons behind their use led to frustration [P42].

In some cases the misunderstanding of agile led to *"doing mini waterfalls"* [P6]. Managers used Scrum terminology, but had the teams commit to unreasonable workloads [P6]. In another case, a waterfall nature was evident as task estimates were given as hours left, and task breakdowns became disconnected from what was really being done [P27].

As a final example of misconceptions, some organizations considered agile being a solution to all problems [P9]. Success was declared prematurely [P49] but expectations created by successful early experiments were not fulfilled [P38].

Lack of guidance from literature. Several cases describe how an agile method was hard to learn from the literature [P21, P27], especially when it comes to implementing it on a large scale [P13]. As Beavers [P5] writes: *"There simply was not a manual or document where we could find easy answers on how to do things"*. Schnitter and Mackert [P45] report that theoretical considerations on how to scale up the agile methods were not good enough, and that product managers and architects struggled when several Scrum teams were working concurrently. Another case concluded that all practices suggested by XP did not fit enterprise scale development, and some practices required customization [P49].

Benefield [P6] describes how it was difficult to find a balance between prescribing a by-the-book implementation which may put people off, and giving too much freedom in the agile methods, which may weaken core practices.

The reality where the practices needed to be applied was described as messy in comparison to the ideal circumstances presented in the literature [P10]. Thus, it was difficult to choose a

method to start with and gain buy-in for [P10]. Difficulties in choosing an initial approach were also due to variances perceived in the agile approaches suggested in literature [P30].

Agile customized poorly. Furthermore, the difficulty of and misunderstandings related to agile were evident in cases where the methods were customized poorly. As a by-the-book implementation often was not feasible, organizations attempted to tailor the agile method to suit their specific needs. However, in some cases this simply meant skipping practices, which led to problems. In one case, certain individuals were allowed to ignore core elements of Scrum, which turned the teams' decision making into a variant of command and control [P9].

In one case [P29], there was a temptation to strip some agile practices and enhance others. Previous attempts had proven that one of the reasons for agile implementations to fail was deviations from the process, because of which the agile mindset did not take root [P29]. A poor customization may lead teams to adopt only practices that reflect their current needs, thus failing to achieve any real change in process and mindset [P9, P46].

A further case of poor customization included replacing the agile vocabulary with familiar terminology from the old approach [P44]. An important benefit of introducing new vocabulary is that it stimulates new ways of thinking [P44].

Reverting to old way of working. In several cases, challenges in the transformation resulted in people reverting to the old way of working. In some cases it was only a temporary struggle while learning agile practices, but in other cases the old way of working displaced agile. Development work has to go on during the transformation but there will be new things to learn for the team. Stress caused by the combination of schedule pressure and much change at once can pull people back to the old way of working [P11, P18, P38].

Even subtle trouble may put the transformation at risk, as people will always look for reasons to revert to familiar behavior [P44]. Teams without adequate training were struggling with applying agile practices correctly, and the challenge the new practices posed made people abandon them and return to the ways they know [P6, P49].

In some cases agile was seen as an overhead, and was therefore abandoned. For instance, as new practices were being introduced there was a decrease in performance, and when teams realized that the benefits are not immediate, they started reverting to the old way of working [P21].

Evans [P12] describes that new senior appointments made management less favorable towards agile, and a waterfall development method started to return.

Excessive enthusiasm. A phenomena that troubled some organizations was over-enthusiasm towards agile methods. Several cases contained reports of change leaders or teams becoming agile zealots. For instance, some members of the agile community could become too evangelic, making people take sides for or against agile development [P12]. Also, while starting out with an overly eager attitude, team members' interest faded, and they reverted to old ways of working when the new approach did not deliver immediate benefits [P20].

Attempting to implement agile too strictly may cause conflicts, and especially when implementing large-scale agile, the change leaders need to maintain a collaborative attitude towards various groups in the surrounding organization [P6]. Introducing agile methods does not guarantee success, and therefore they should not be followed blindly [P4].

4.2.4. Coordination challenges in multi-team environments

Interfacing between teams difficult. One of the most prominent transformation challenges was the difficulty in coordinating the work of several agile teams.

Challenges arose when teams needed to work with other teams, and as parts of the larger surrounding organization [P13, P17, P24]. While introducing agile had created flexibility at the team level, the surrounding organization was not responsive enough [P26]. The roll-out of agile had not removed dependencies, and the dependencies made managing development difficult [P9, P10].

Autonomous team model challenging. Some organizations initially created models in which teams operate autonomously — well in line with agile principles. However, a number of issues arose from this independence. For instance, teams needed to strike a balance between the their own goals and the broader goals of the organization, but often chose to focus only on their own goals [P7]. Coordination was obstructed by independent teams that did not respect the larger context [P13]. Coordination was also difficult when teams were working independently for different customers, but the applications being built were interdependent [P29]. One case reported that even allowing teams to have different Sprint durations created delays in delivery [P51].

Global distribution challenges. Further coordination problems were encountered when scaling up agile over many geographically distributed sites. Distribution had negative effects, such as missing kick-off meetings, reduced feelings of proximity when telecommunication is necessary, and difficulty in arranging frequent meetings due to time zone differences [P45]. Agile project management was also problematic. In a waterfall way of working, separate parts of projects could be isolated to different sites, but the agile way of working does not allow such strict slicing of projects [P29]. In one case, it was simply admitted that a distributed organization will impose additional burden on communication and require additional care in the process, but distribution and agile would still be used together [P48].

Achieving technical consistency. Technical problems relating to inter-team coordination were reported as well. Integrating the products of teams was problematic in some cases [P5, P50], and the lack of standardized build scripts in one case [P27]. There were also problems in synchronizing the definitions of software interfaces between teams, and dependencies in code caused problems in larger features [P26]. One case reported that the strong focus on individual teams in the agile way of working created a fragile architecture, divergence in coding style, and even distrust between teams [P24].

One case describes the progression of coordination problems. First, an attempt was made to reduce cross-team dependencies, but it became evident that the dependencies were an inherent part of the project. A traditional approach to managing dependencies created excess work, reduced independence and flexibility of teams, and created contract-based and adversarial relationships. Coordinating using Scrum-of-Scrums worked for up to five teams, but did not scale to a global level, as teams were focused on their own work and the new practices narrowed communication channels and created communication breakdowns. It was concluded that a balance is needed between completing new stories from the team backlog and maintaining overall stability of the application. [P34]

4.2.5. Different approaches emerge in a multi-team environment

Interpretation of agile differs between teams. When many teams implement agile without consistent guidance, friction and fragmentation may emerge [P9]. The organization may require moving people between teams from time to time, and therefore it is desirable that the agile cultures of different teams are not too different. Divergence in process creates increased costs when relocating people [P38]. Further, forecasting and benchmarking teams become difficult [P38, P48]. To overcome problems with divergence in agile approaches some organizations defined standards [P8, P43].

Using old and new methods side by side. In most cases, the organizational transformation proceeded gradually, and during the process it was possible that the new agile method was used in parallel with the old methods. Using the old and new methods side by side was generally seen as problematic [P1, P26], causing tensions on all organizational levels [P10]. Ongoing projects had been set up with old development methods, and the agile methods needed to be arranged to fit in with them [P8, P29]. A particular problem in collaboration between waterfall and agile projects was that in agile, the software design was fleshed out over time as sprints progressed, but the waterfall method required a detailed design to be frozen upfront [P29].

4.2.6. Hierarchical management and organizational boundaries

Middle managers' new role in agile unclear. The need for additional management positions in a larger organization may pose problems to agile processes that emphasize self-organization. Especially the role of middle management was unclear in agile methods [P2]. This is problematic, as an agile transformation requires a cultural change particularly on the middle management level [P2]. Managers were reported to need to resist the tendency to command and control and allow room for self-organization, but the change in mindset was difficult to achieve for the people involved [P11, P52]. One case [P30] describes how the project management group had previously worked through big up front plans and competed for resources, but those ways would need to dramatically change.

Several other problems related to management roles were also presented. For instance, [P49] describes how managers were left outside the roles offered by the new agile way of working. In another case, when managers were appointed as Scrum masters, developers felt being micromanaged [P27]. This was partially because of a poor understanding of the agile method [P27]. It was also described how mixing the role of project manager and Scrum master created a conflict of interest, and turned the Scrum manager's role into one of policing instead of supporting the team [P30]. In some cases, problems with manager positions were solved by phasing out roles relating to the old way of working, and replacing them with new positions more suitable for agile development [P38, P52].

Management in waterfall mode. Even after adopting agile, there were cases where management continued to work according to the old waterfall model. One case [P38] described management as *"focused on meetings and big up-front project plans"*, despite having adopted agile. In another case, management was losing confidence in agile because reports on costs and progress were not produced in the same format as before [P35]. As Scrum teams did not commit to fixed schedules, they were considered unreliable [P45].

Some cases reported that development efforts were still controlled on the top level by a project management office (PMO). The PMO was described as entrenched in a waterfall paradigm, and the top-level rigidity was causing friction in the agile adoption [P3, P6]. The PMO was seen as a hub and bottleneck controlling all aspects of projects, and its central structure had to be dismantled for the organization to become agile [P46].

Keeping the old bureaucracy. There were also problems with duplicating bureaucracy when two different ways of working were in effect. For instance, agile teams were required to comply with current procedures producing excess documentation and stepping through approval gates [P20, P21]. The bureaucracy of the previous traditional model was still enforced, and management was not willing to lighten the process. Another case describes that after introducing the agile method teams were required to fill in templates of two processes [P48].

Internal silos kept. In some cases, the initial organization had internal boundaries, or specialized knowledge in silos, causing problems in the agile implementation. Cloke [P10] describes how the use of Scrum revealed an internal segmentation where teams operated with differing priorities and agendas. This hampered the initial effort to use Scrum in a larger context [P10]. Cowan [P11] describes how projects needed specialized skills that were scarce and people were often relocated to match the needs of the moment. Too much reorganization made it difficult for teams to plan ahead [P11].

4.2.7. Requirements engineering challenges

High-level requirements management largely missing in agile. While some agile methods, e.g. DSDM, have a quite structured approach to requirements, e.g. Scrum which was the most widely adopted one does not, nor does, e.g. XP, another widely used approach. Large development projects demand high-level requirements management. This is apparent in cases where product complexity requires additional layers of product management [P45], requirements are created by several stakeholder groups and development teams cannot be in contact with all of them [P39], or do not have access to stakeholders due to distribution [P24].

Requirement refinement challenging. In some cases, requirements were initially defined at a high level in marketing requirements documents [P1, P31] or functional specifications [P33]. These high level requirements needed to be elaborated to be useful by agile development teams. The requirements refinement in itself [P1, P33], when to do it, and to which level of detail [P17] were all reported as challenges.

Creating and estimating user stories hard. Product managers and business analysts struggled with creating high level requirements [P33, P37, P51], and teams struggled breaking them down to a size that is possible to do effort estimation on [P33]. One study [P48] describes how high level product management delivered requirements in large chunks, resulting in development teams spending huge amounts of time defining suitably sized stories.

Several cases highlighted that much learning was needed to master the new process of creating user stories, both on product management and development levels [P27, P30, P33, P37]. There were problems such as ambiguity in requirements [P5, P11], and effort estimation for stories [P27, P37].

Gap between long and short term planning. High level requirements work and estimation was problematic as there was a gap between short and long term planning [P9, P10]. Typical agile backlogs give only short-term visibility, creating a need for additional practices for long-term planning [P26, P31].

Several cases described that care had to be taken to avoid long term planning becoming a prescriptive practice. In order to preserve the agility of development, a schedule-driven approach was avoided [P10], sensitivity was used when considering setting milestones [P13], and striving for more accurate estimates was not allowed to become an excuse for gathering requirements up front [P38].

4.2.8. Quality assurance challenges

Similarly to requirements engineering, the agile approach may need to be extended to accommodate additional testing activities [P31]. Organizations may have existing separate QA teams that must be coordinated to work with development teams [P17, P30, P48].

Accommodating non-functional testing. Full quality assurance of a system might require special testing such as performance, load, and memory testing, but agile methods lack a focus on them [P17]. These testing activities can not be done within the boundaries of user stories, and require more resources than teams can spare [P31]. When testing tasks overlap team boundaries, it is sensible to have separate teams for the tasks, but coordination between specialized QA teams and development teams must be defined [P28, P48].

Lack of automated testing. Several cases reported problems with lack of automated testing. The lack of automated tests caused excess testing work and late discovery of defects [P10, P11, P17].

Requirements ambiguity affects QA. Another challenge in QA was the relationship to requirements engineering. The QA functions were struggling if there were problems such as ambiguity or ineffective breakdown of large requirements [P48]. In the waterfall way of working, an extended time was reserved for QA at the end of projects, which allowed retroactively resolving ambiguities in requirements [P5]. However, ambiguity became an impediment to QA when development worked in short increments and there was no extended period for stabilization [P5].

4.2.9. Integrating non-development functions in the transformation

Several cases described how introducing agile started from the development teams [P2, P17, P31, P38]. However, development operates in a larger context and needs to interface with other organizational functions, causing challenges.

Other functions unwilling to change. Challenges were faced when exposing other parts of the organization to the agile way of working, as functions distanced from development were resistant to change [P2, P18, P31]. Tension grew between development and other roles in the organization [P1]. The full benefits of the transformation could not be attained unless the entire organization was set to work along the same paradigm [P18, P42].

Our primary sources listed various functions beyond development that interface with development and are affected by the agile transformation. Marketing was particularly frequently mentioned [P5, P17, P28, P31, P38, P45, P48]. Other groups included sales [P17, P31], infrastructure and operations [P10, P45, P46, P50], user experience and design [P9, P10, P28], documentation [P28], legal [P10], security [P10], customer services [P48], and finance [P48].

Challenges in adjusting to incremental delivery pace. The iterative nature and the fact that agile affects timings in the software delivery life cycle, often caused problems in the interface between development and other functions.

Iterativity changed the pace of delivery [P17], and forced design to focus on a shorter scope [P15]. Various groups feeding and supporting development had earlier committed once to a long term plan, but in the new arrangement they needed to adjust to incremental deliveries at a faster pace [P48].

Especially user experience (UX) and interaction designers did not welcome an incremental model [P6], and were struggling with maintaining the big picture in design while working in iterations [P9, P10]. The infrastructure and operations teams fell behind due to the increased speed of development teams, and were forced to update their way of working [P46, P50].

Federoff and Courage [P15] elaborate on the user experience team's problems with the short time horizons of Scrum. Previously, the UX team had supplied development with designs within a long release cycle, but Scrum required completing features within weeks. The problem was that the UX work required a holistic view instead of an incremental one, and the design process did not fit well into the time frame of a sprint. The UX team was overloaded and discontent, but the situation was fixed by refining the schedule for interactions with the development teams. [P15, P28]

Challenges in adjusting product launch activities. A second type of timing problem was that certain release activities inevitably have long lead times, and required the commitment to a set of functionality early on. Agile's emphasis on short time horizons and flexibility in prioritization did not mix well with such activities. For instance, preparing printed marketing material and press releases requires information to accurately present the upcoming features of the product [P38].

One case describes that marketing needed three months for preparing the product launch, but the agile process allowed development to keep changing the content of the product. As a result, marketing struggled, creating material and campaigns without knowing the exact product features. Further, the processes of acquiring export clearances and licensing authority required that the feature set of the product was known well in advance. Development felt that these requests slowed them down [P31].

Rewarding model not teamwork centric. In several cases, human resources (HR) was mentioned as working against the agile adoption. In order to gain the full benefits of going agile, the entire organization should to be aligned, including HR [P38, P42, P49]. Especially the rewarding practices set by HR were seen as problematic, as rewards were tied to personal performance, acting against the team-centric thinking and the agile approach in general [P3, P6, P49]. One case even reported a practice of tracing failures down to individuals [P40].

4.3. Success factors in transformation

In this section answer our second research question RQ2: *What success factors have been reported for large-scale agile transformations?* We organized the 29 success factors, each identified in several primary studies, into eleven categories, which are elaborated in this section and summarized in Table 12.

4.3.1. Management support

Ensure management support. Numerous cases made it clear that management support is an absolute necessity [P8, P18, P27, P31, P33, P36, P43, P47, P49]. This was reflected in statements such as *"Adopting agile, or implementing any significant change, requires an executives sincere support."* [P44], *"Executive commitment was crucial to implementing massive change."* [P16], and *"Having upper management engaged, supportive and visible is critical for wholesale organization involvement with Scrum."* [P11].

Managers were seen to be in a key role in making the change stick, as they had the authority and power to remove impediments [P18]. Seffernick [P46] describes how a number of people attempted to explain away the applicability of agile methods, but the objections were overruled by the director's commitment to make agile work. Management support was similarly needed when tight release schedules had to be flexed in order to give room for the adoption process [P16, P27]. Favorable management decisions were also critical when additional resources were allocated to training and coaching [P47].

Make management support visible. Visible involvement of management was reported to motivate and encourage employees to adopt the new way of working [P40]. For instance, the CTO organized training sessions personally [P24], and the engineering VP frequently visited sprint demos [P11]. When the corporate level support for the agile initiative was showing teams adopted agile methods even spontaneously [P9].

Educate management on agile. In order to gain support for agile several cases highlighted the need to educate management [P8, P47]. Proper education ensured that managers would not enter a command and control mode, which would harm the agile implementation [P46]. Managers not understanding the principles of agile felt left out with the introduction of self-organizing teams, which sometimes resulted in backlashes [P6]. Providing proper training corrected the situation, and even created strong agile supporters in management [P6, P9]. Training cleared misconceptions [P29], and helped create a consistent implementation of the agile approach across the organization [P11].

4.3.2. Commitment to change

Communicate that change is non-negotiable. Feedback from the personnel should certainly be listened to, but in the end it

Table 12
Success factors for large-scale agile transformations.

Success factors	Primary sources	Case organizations	# of cases
Management support			**16 (38%)**
Ensure management support	P8, P11, P16, P18, P27, P31, P33, P36, P43, P44, P46, P47, P49	C7, C9, C11, C13, C21, C24, C27, C33, C34, C36, C37, C39	12 (29%)
Make management support visible	P9, P11, P24, P40	C5, C9, C18, C31	4 (12%)
Educate management on agile	P6, P8, P9, P11, P29, P46, P47	C5, C7, C9, C22, C36, C37	6 (14)
Commitment to change			**7 (17%)**
Communicate that change is non-negotiable	P11, P22, P48, P49	C9, C16, C38, C39	4 (10%)
Show strong commitment	P8, P10, P11, P43	C7, C8, C9, C33	4 (10%)
Leadership			**7 (17%)**
Recognize the importance of change leaders	P3, P4, P11, P16, P18, P31, P33, P42	C2, C3, C9, C11, C13, C16, C24	7 (17%)
Engage change leaders without baggage of the past	P11, P16	C9, C11	2 (5%)
Choosing and customizing the agile approach			**20 (48%)**
Customize the agile approach carefully	P8, P10, P11, P29, P31, P40, P41, P43, P45, P49, P51	C7, C8, C9, C22, C24, C31, C32, C33, C35, C39, C41	11 (26%)
Conform to a single approach	P8, P11, P12, P16, P32, P43	C7, C9, C10, C11, C17, C33	6 (14%)
Map to old way of working to ease adaptation	P3, P14, P16, P29, P44, P48	C2, C11, C12, C22, C34, C38	6 (14%)
Keep it simple	P5, P10, P16, P40	C4, C8, C11, C30	4 (10%)
Piloting			**14 (33%)**
Start with a pilot to gain acceptance	P3, P9, P14, P17, P31, P36, P38, P39, P47	C2, C4, C5, C12, C24, C27, C28, C29, C37	9 (21%)
Gather insights from a pilot	P8, P22, P27, P40, P45	C7, C16, C21, C31, C35	5 (12%)
Training and coaching			**15 (36%)**
Provide training on agile methods	P3, P6, P14, P16, P30, P37, P46	C2, C5, C11, C12, C23, C28, C36	7 (17%)
Coach teams as they learn by doing	P3, P6, P8, P9, P16, P17, P18, P29, P30, P33, P42, P44, P47, P48	C2, C4, C5, C7, C11, C13, C16, C22, C23, C34, C37, C38	12 (29%)
Engaging people			**9 (12%)**
Start with agile supporters	P6, P29, P34	C5, C22, C25	3 (7%)
Include persons with previous agile experience	P6, P10, P37	C5, C8, C28	3 (7%)
Engage everyone in the organization	P4, P6, P16, P22, P31	C3, C5, C11, C16, C24	5 (12%)
Communication and transparency			**7 (17%)**
Communicate the change intensively	P16, P22, P33, P40, P41, P43, P48	C11, C16, C31, C33, C38	5 (12%)
Make the change transparent	P6, P11, P16, P42, P43	C5, C9, C11, C16, C33	5 (12%)
Create and communicate positive experiences in the beginning	P6, P32, P29, P40, P42, P46	C5, C16, C17, C22, C31, C36	6 (14%)
Mindset and Alignment			**17 (40%)**
Concentrate on agile values	P1, P16, P42, P46, P48	C1, C11, C16, C36, C38	5 (12%)
Arrange social events	P11, P23, P27, P32, P40, P41	C9, C17, C21, C31, C32	5 (12%)
Cherish agile communities	P3, P12, P41, P42, P47	C2, C10, C16, C32, C37	5 (12%)
Align the organization	P8, P18, P31, P42, P43, P46	C7, C13, C16, C24, C33, C36	6 (14%)
Team autonomy			**10 (24%)**
Allow teams to self-organize	P5, P16, P18, P27, P30, P34, P41, P45	C4, C11, C13, C21, C23, C25, C32, C35	8 (19%)
Allow grass roots level empowerment	P3, P6, P41	C2, C5, C32	3 (7%)
Requirements management			**10 (24%)**
Recognize the importance of the Product Owner role	P12, P14, P16, P24, P30, P33, P39, P46	C10, C11, C12, C18, C23, C30, C36	7 (17%)
Invest in learning to refine the requirements	P5, P11, P17, P33, P48	C4, C9, C11, C38	4 (10%)

must be made clear that the old way of working can not be returned to [P22, P49].

For a case where the organization was changed all at once it was reported that the magnitude of change helped to create an impression that the change was non-negotiable [P11]. Firstly, as it was evident that a large change was to happen people felt encouragement and permission to move on from the old way, and secondly, the urgency eliminated wasteful discussions on whether Scrum was a good method or not [P11].

When the organization culture is ingrown, the change vision must constantly be reminded of, and each step towards the new way of working considered as a victory [P48].

Transformation can be facilitated by setting up mechanisms that force change. Spayd [P49] describes that senior management pushed hard on a mandate to have deliveries every 90 days. The mandate made change necessary, and while the strong pressure did not promote agile practices in every case, the drive for change was perceived good in general [P49].

Show strong commitment. A large change will inevitably require strong commitment, but problems in the transformation can put the commitment to test. The agile approach is introduced because of problems in the old way of working, and therefore there will be organizational issues uncovered during the transformation [P10]. People must not be demoralized when facing challenges, and the determination to change must be maintained [P8, P10]. Problems might not be due to the agile approach — in some cases it can expose existing problems in the organization [P10].

Ryan and Scudiere [P43] describe that management commitment was showing as a strong focus on certain agile practices that had been chosen as non-negotiable, and constant assessment that the practices work. The expectations were clearly communicated to the teams, and constant assessment made it clear that change was desired [P43]. A strong commitment from management assures teams that the change is the right thing to do [P11].

4.3.3. Leadership

Recognize the importance of change leaders. Transforming the way of working of a large group requires coordination and leadership [P31, P42]. In addition to the leadership provided by coaches, specific change leaders were mentioned. Cases indicated the importance of having spokespersons for the change [P3, P4, P31]. Cowan [P11] describes how one person was strongly driving the transformation, and made an indisputable contribution for transforming the organization. Maples [P31] describes how the change and agile adoption was guided by one “counselor” [P31]. Other cases described that the change was led by a competent roll-out team, which had representatives from all parts of the organization [P16, P33]. Finally, the responsibility of line and project managers to act as change leaders was highlighted [P18].

Engage change leaders without baggage of the past. A few cases discussed the benefit of having change leaders without baggage from the past. Cowan [P11] describes how the newly hired director was able to circumvent territorial battles that existed from before, and therefore focus fully on getting the agile approach implemented. In another case, external coaches were described to better spot places for correction in the agile approach, and their advice was received better because they were considered impartial when being external to the organization [P16].

4.3.4. Choosing and customizing the agile approach

Customize the agile approach carefully. Customizing the agile approach and practices was often seen as a necessary step in the agile implementation. As each organization will have its own challenges in adopting agile, it should be carefully considered which organization specific areas to focus on when choosing the agile practices to implement [P8, P43]. Lewis and Nehrer [P29] recommended choosing the agile approach according to the current corporate model in order to avoid interference.

Cases reported that customization was part of a successful agile implementation. For instance, teams were let to customize their practices individually, to fit the needs of each team [P41]. Spayd [P49] writes that teams who modified agile practices to fit the large and distributed environment ended up doing better than teams that did not.

To allow teams to become innovative and perform well, the agile approach should be customized in a pragmatic way instead of following a strict textbook interpretation [P10]. In order to draw the most out of agile, teams should innovate and find their own practices that really work for their case [P40, P41].

Especially in a larger organization it is not feasible to use the same process for all projects [P49]. An individual application of the agile process is needed for different types of development, depending on, e.g., the type of software being developed or the project size [P45]. Applying agile at scale will require to deviate from some of the typical recommendations [P31]. However, it is essential to remember the agile principles when doing customizations, and watch out for customization that would contradict the principles [P31].

The customization of the agile approach was also seen as an evolutionary process. Cowan [P11] described that in the big bang transformation it was useful to selectively compromise some parts of the agile implementation, so that the core practices were firmly set in place. The agile transformation is a constantly ongoing process, and it is recommended to regularly challenge teams to refine the agile implementation to reflect the current needs of the organization [P41, P43].

Conform to a single approach. Studies reported that conformity to a single approach should be considered when implementing agile. A consistent common vocabulary was seen to benefit the organization and support the change [P8, P16, P43]. Other benefits of using a single approach were the possibility to compare work between teams, easier relocation of people, and predictable progress for the stakeholders [P43]. Consistency and common understanding was created in meetings with peers in the same job discipline, in public events, and by peer coaching [P11, P12, P32].

Map to old way of working to ease adaptation. Mapping the agile approach to the old way of working was seen as necessary in a few cases. Although a general mapping to the old way of may not be good [P44, P46], some cases needed to preserve high-level management practices [P14, P29, P48]. By allowing management to remain unchanged it was possible to introduce agile step-wise on the team level, which helped in getting management buy-in for the process [P14, P29, P48].

In one case, Scrum was considered as a wrapper for existing practices, and could therefore easily fit in the organization [P44]. The agile methodology was considered complementing the existing culture, which eased management buy-in for the new method [P44].

A few organizations had structures in place that were similar to agile, making the transformation easier. For instance, a previous organizational model based on small and autonomous teams strongly aided the adoption [P3]. Another case described the original on-demand delivery model a natural fit for agile, and the transformation was presented as a return to core values instead of a remodeling [P16].

Keep it simple. One piece of advice given by a few cases was to keep the organization and process simple [P10, P16]. Beavers [P5] describes that attempting to operate in a complex and global setting complicated even simple agile practices. The organizational chart was simplified, which was welcomed by both employees and managers [P5]. Rather than focusing on details in process, commu-

nication practices, and tools the focus should be on engaging the people in the teams, and forming the process based on what can be seen to work in practice [P40].

4.3.5. Piloting

Start with a pilot to gain acceptance. Having a pilot project was reported as a significant success factor in several cases [P3, P9, P38]. The pilot project helped create confidence that the agile way of working would be suitable for the organization and the general acceptance of agile was increased [P3, P17]. The pilot also cleared disbeliefs of the suitability of large-scale agile, and created acceptance both in development and management [P9, P31, P36].

Pilots were especially important for gaining management acceptance. Cases reported that management gave approval for agile methods only after seeing successful pilots [P14, P47]. Prokhorenko [P39] describes that seeing the example of successful pilots will make managers from other departments eager to try the new method, and thus help to spread its use.

Gather insights from a pilot. A benefit in piloting was that it provided knowledge on how to fit the agile method to the particular organization. A pilot project enabled the organization to get feedback on how teams and managers are best introduced to the new methods [P8, P22]. Organizations met challenges in the pilot projects. However, the pilot projects served as valuable learning experiments providing insights on how to mitigate problems when the rest of the organization is transforming [P27, P40, P45].

4.3.6. Training and coaching

Provide training on agile methods. Several studies stated that training improved the chances of succeeding in the transformation [P3, P16, P30]. It was even highlighted that the change would have failed without training [P14], as Benefield [P6] claimed: *"we saw again and again how training could make the difference between success and failure for a team"*. Training also helped people to become more positively inclined towards the new way of working [P16, P37], and at best training made people enthusiastic to change [P3, P46].

Coach teams as they learn by doing. Agile methods avoid prescribing exact ways of working, and rather emphasize a mindset of adapting to the situation. It is difficult to explain by theory how such a mindset should be applied, and the agile practices are best learned by doing. Coaching teams while they apply the agile methods in practice was seen as an important factor in change [P3, P16, P17, P30, P47]. A few cases stated even that coaching was essential for success in transformation [P8, P18, P29, P33]. Also, without coaching, teams can do damage to the agile transformation if techniques are applied improperly [P9].

There were a number of statements on the benefits of coaching. A coach can watch for and correct problems when they arise [P16, P42]. Coaches helped draw attention away from a focus on tools towards understanding the principles of agile [P42, P48]. There were also benefits in using both internal and external coaches. External coaches were able to have an objective view of the organization [P16, P44], whereas internal coaches were more accessible and knew the specifics of the organization [P3, P6].

4.3.7. Engaging people

Start with agile supporters. Choosing people with a disposition towards agile methods was seen as a requirement for the change to take the right track. Teams are staffed usually based on technical competencies, but also considering personality aspects was emphasized for agile teams [P29, P34]. The personality of people was seen as a key aspect for achieving change [P6]. There is a need for collaborative and understanding persons who are prepared to discard preconceptions and willing to try new untested approaches [P6, P29].

Include persons with previous agile experience. A few cases mentioned the importance people with previous agile experience had at the start of the transformation [P6]. For instance, it helped the product management office to implement agile over the entire enterprise when the staffing was updated to include people with agile experience [P10]. Staffing teams partially with developers familiar with agile helped the rest of the team get a good hold of agile development [P37].

Engage everyone. Many cases highlighted the importance of engaging people broadly in the organization. One of the lessons learned presented was that it is important to involve all stakeholders to gain acceptance of the transformation [P4, P31]. The transition team made an effort to engage people by both inviting them to give feedback online and by holding an extensive number of feedback meetings [P6, P16, P22]. Being inclusive towards everyone was seen as one of the key success factors in the transformation [P16]. Inclusiveness will encourage people to participate and visibly work in the new agile way [P16].

4.3.8. Communication and transparency

Communicate the change intensively. Intensive communication was emphasized in a number of studies. It is important to reach as many people throughout the organization as possible, as without communication the new way of working will not take root [P16, P43]. It was recommended that working in the new agile way is made highly visible on many communication channels and even over-communicated [P16, P33, P41]. Mencke [P33] summarizes the viewpoint on over-communication: *"Even if you think that your teams understand a new method or process, repeat your communication to be sure."* Workshops, coaching sessions, online discussions, and newsletters are examples on different communication formats used [P41]. Another approach in communicating the change was that managers explained and encouraged change in an extensive series of one-on-one discussions [P22].

Some studies emphasized communicating the goals of the transformation. Having a clear message of expectations helped remove confusion and make people understand the purpose of the transformation [P43, P48]. The motivation to change can be increased by having a simple proposal on how to reach the desired outcomes [P40]. McDowell and Dourambeis [P32] describe how the organization launched a series of communication events especially designed to emphasize the reasons behind the agile practices.

Make the change transparent. Enabling transparency during the transformation was reported as important, and even highlighted as a critical factor for success [P11, P16]. Fry and Greene [P16] underline the importance of transparency and sharing of information: *"This bias to sharing information with everyone was critical in our ability to adapt on a daily basis to ensure our success."* Transparency was achieved by showing both successes and challenges [P6], actively reaching out for feedback [P6], using project management tools to display project status publicly [P11], by rearranging physical spaces [P42], and by holding the meetings of the roll-out team in public [P16]. By sharing experiences and status of the transformation the organization was aligned and everyone was moving in the same direction [P42, P43].

Create and communicate positive experiences in the beginning. Several cases highlighted that the agile transformation spread effectively through positive word-of-mouth [P6, P40]. The move towards agile is assisted by making any benefits publicly visible and celebrating even the small victories [P40, P42]. When good results were shown by a team it created interest in others, and enthusiasm to try the new way of working would spread [P39, P46]. Some companies used agile and waterfall methods side by side. This setting made comparison possible, bringing up the benefits of agile [P19, P36].

Although a recipe to fabricate positive experiences can not be given, many cases described how they had succeeded. Benefield [P6] describes how a survey was done to objectively measure satisfaction in agile, and the positive results helped teams make the decision to change their way of working. Further, management requested proof on better performance as a response to requests to increase coaching and training budget. After conducting inquiries, the coaching team was able to present an estimate of a 30% increase in performance, which made management highly convinced [P6].

4.3.9. Mindset and alignment

Concentrate on agile values. In a number of studies it was described that the principles of agile should be emphasized over practices and simple mechanics [P46, P48]. When people understand the agile values they will also understand why the change is being done and feel motivated [P1, P16, P42].

Some cases described that inexperienced coaches and Scrum masters made mistakes in focusing too much on the implementation of practices [P23, P30].

Arrange social events. Social events were reported to benefit the transformation by helping to build an agile mindset. A few cases even described the transformation being driven by a series of events where people received information and had the possibility to participate in shaping the new way of working [P32, P41]. One case described how the corporate agile awareness and teaching event was designed to be fun and engaging, which made people more enthusiastic in applying the agile practices [P32].

Various social activities were presented as valuable tools for improving bonding within teams [P11, P27, P40]. The importance of creating personal bonding was also highlighted for change leading entities such as management and coaching teams [P11, P23].

Cherish agile communities. The formation and influence of agile communities was reported to have a significant impact on transformation. The emergence of an agile community was reported as a success factor in the transformation [P3]. A community was also seen as indispensable as it has power to overcome impediments that would be too large for individuals to affect [P47]. The agile communities raised awareness of agile methods in the organization [P41, P42, P47], and spawned eager followers who spread the word even further [P12].

Align the organization. A factor in achieving change was creating alignment towards the common goal of introducing new development methods. It was seen as an important factor that all levels in the organization speak the same language and accept the change [P18, P31]. Alignment was built by promoting success stories and gathering lists of problems to tackle [P42, P46]. In one case, a complete alignment between teams and within management was considered as necessary in order to use agile in a large context [P43]. Another case highlighted creating and applying a structured roll-out process to uniformly introduce agile to a large number of teams [P8].

4.3.10. Team autonomy

Allow teams to self-organize. The agile principle of giving teams the power to decide over themselves was seen as an important factor in advancing the transformation. In some cases, management initially attempted to prescribe how the new practices should be implemented [P5, P18, P41]. However, it was learned that only when full control was given to teams the agile methods could be properly established [P5, P16, P41]. Allowing self-organization creates commitment to the change and motivation to continued use [P16, P18]. It allows teams to take ownership of the development process and voluntarily improve it even further [P41].

The acceptance of agile methods was easy when teams gained the authority to decide on development speed and quality [P45]. Favoring empowerment over prescribing the new practices was also reported to improve performance [P27]. Giving teams the authority to decide over work items increased productivity and morale [P30, P34].

Roche and Vasquez-McCall [P41] describe how prescribing the agile methods to use led only so-far in the transformation. The effectiveness of teaching and communicating the transformation started to decline. To enable the transformation to proceed further, an organization-wide challenge was created where teams would independently develop and showcase the best agile practices. As a result the ownership of the methods was transferred to teams, and the new way of working gained a secure foothold [P41].

Allow grass roots level empowerment. An interesting success factor was the absence of a top-down mandate. Atlas [P3] describes that the use of Scrum was spreading because teams were both allowed to use and enabled to learn the method. The transformation itself was agile when people felt empowerment in making the decision to adopt. The incentive to change was created when teams learned about agile methods, and perceived that a change would be beneficial [P3]. The absence of mandate granted genuine support for the new way of working on the grass roots level, which was a necessity for the process to work [P6]. It was also thought that proceeding with a top-down mandate would have made teams conform to a single process [P41]. This would have been sub-optimal, as it was important that each individual team and project tailored their practices [P41].

4.3.11. Requirements management

Recognize the importance of the product owner role. A particularly important role was the Product Owner. Many cases reported problems if the Product Owner did not perform adequately, and it was seen as critical to have a dedicated person in that role [P12, P14, P30, P39]. Successes or problems emerged respectively depending on how well the Product Owners were performing.

In one case, a well-implemented Product Owner role was understood to be a key success factor when using agile methods [P16, P33]. It was reported that when the Product Owner role was implemented correctly, the team performed better and the work products were correct [P24]. Projects started off better when the Product Owners were engaged early on [P46]. Some Product Owners were even described to become agile enthusiasts when they learned how business and technology can collaborate in the agile way of working [P46].

Several cases endorsed training and coaching for the Product Owner role [P16, P39, P46]. Product Owners should receive training on agile principles, backlog management, user story breakdown, and agile planning [P16]. Also tool support for backlog management and two way communication between Product Owners and teams should be worked on [P39].

Invest in learning to refine the requirements. Some cases reported challenges in requirements management, highlighting the difficulty in managing the gap between high-level requirements and user stories handled by teams. It is important that the requirements are concise and small enough for the teams to handle [P5, P11]. It was recommended to invest in teaching skills in breaking down and writing user stories [P11, P33, P48]. Gat [P17] describes how the gap between development and requirements management was bridged by having a dedicated "requirements architect" to help with requirements refinement.

5. Discussion

In this section we first discuss our general observations, followed by a discussion on the answers of our research questions. Then, we identify discrepancies and open issues in the literature.

Finally, we discuss the limitations and present a future research agenda.

5.1. General observations

In the past decade several publications have discussed the use of large-scale agile. We identified 52 papers presenting 42 industry cases describing large-scale agile transformations. The identified primary studies were almost exclusively experience reports. We could only find six research papers that we classified as case studies. None of these papers had their research focus directly on the transformation process, even though they provided some information on the transformation process as well. As can be seen from our relevance classification in Table 12. four research papers received classification 2, while one received 3 and one 1 (with scale 1–5, where 5 means that the entire paper focuses on describing the transformation process). When comparing to experience reports, where 20 of 46 papers received 4 or 5 in our classification, we can conclude that experience reports concentrated better on our research topic, lager-scale agile transformations, than the research papers.

Moreover, as the second most often mentioned challenge in our primary studies rose the "Lack of guidance from literature" which was mentioned by nine cases.

Thus, the main finding of this study is that despite the relevance of the topic for practitioners, research is seriously lagging behind, and there is in particular a need for rigorous case studies and summarizing research.

5.2. Answers to research questions

This section summarizes and discusses the answers to the research questions.

As an answer to RQ1: "What challenges have been reported for large-scale agile transformations?", we identified 35 challenges grouped into nine categories, summarized in Table 11. The challenge categories that received the most mentions were: 1) Agile difficult to implement (mentioned by 48% of the cases), 2) Integrating non-development functions (43%), 3–4) Change resistance (38%) and Requirements engineering challenges (38%). From individual challenges the most mentions received: Other functions unwilling to change (31%) and Lack of guidance from literature (21%).

These results show that large-scale agile seems to be harder to implement than people expect, as companies complain about not finding enough guidance in the literature. With increasing organization size organizational functions beyond development get involved, and they need to interface with development. Such functions range from marketing and sales to user experience and human resources. If these functions are not aligned with the transformation, that might cause serious limitations for the agile implementation and thus the full potential of the agile cannot be realized.

To answer RQ2: "What success factors have been reported for large-scale agile transformations?" we identified 29 success factors grouped into eleven categories, as presented in Table 12. The success factor categories that stand out are: 1) Choosing and customizing the agile approach (50%), 2–3) Management support (40%) and Mindset and Alignment (40%), and 4) Training and coaching (38%). From individual success factors the most often mentioned were: Ensure management support (29%), Coach teams as they learn by doing (29%), Customize the agile approach carefully (26%) and Start with a pilot to gain acceptance.

These success factors show that large-scale agile cannot be just taken into use off-the-shelf, but careful customization is needed to make a good fit for the organization and to gain the best benefits. Even though agile is often seen as a grass-root movement starting from development, management support is clearly needed when performing a large-scale agile transformation. Mindset and alignment rose high as well, emphasizing the fact that understanding the agile values behind the agile practices is important and aligning the whole organization towards the common goals makes it easier to pull through the transformation.

There does not exist similar literature studies on large-scale agile as ours, nor does there exist any surveys specifically on large-scale agile. The studies and surveys that do exist have studied agile in general, not specifically as large-scale nor agile transformations, e.g. Chow and Cao (2008) studied success factors in agile software projects in general. The only studies that touch the topic of this paper are not scientific but done by agile consulting or tool companies, e.g. Version One's State of Agile surveys (VersionOne, Inc, 2016) or Forrester's surveys (For, 2012; Giudice et al., 2014).

The closest to our study is the State of Agile survey, as large part of the respondents of their latest survey (VersionOne, Inc, 2016) were from large organizations that had at least partially adopted agile. The results of that survey have similarities with ours. Their respondents reported as the biggest barriers for further agile adoption the following: the ability to change the organizational culture (55%), general organizational resistance to change (42%), pre-existing rigid/waterfall framework (40%), not enough personnel with the necessary agile experience (39%) and management support (38%). Change resistance rose as a big challenge in our study as well, and we recognized management support as one of the top success factors. Pre-existing rigid waterfall frameworks did not rise as a separate challenge in our study, but we recognized "management in waterfall" as a challenge as well as "using old and new approaches side by side", where the old approach was most often just waterfall or waterfall type of rigid method. The ability to change the organizational culture was not recognized by our study as such, however, challenges close to that were, e.g. "management unwilling to change", "keeping the old bureaucracy" and "other functions unwilling to change". However, as a success factor category we had "Mindset and alignment", which included success factors such as "concentrate on agile values" and "align the organization". Thus, our study recognized some factors that relate to the challenges and the importance of changing the organizational culture. Finally, the barrier "not enough personnel with the necessary agile experience" was not recognized as a separate challenge in our study, but "training and coaching" was among our most important success factor categories.

The State of Agile survey did not bring up our two biggest challenge categories "Agile difficult to implement" and "Integrating non-development functions". One explanation to that might be that the survey most probably had pre-defined answer categories limiting answers to those, whereas our categories rose from the data, and thus were not limited to any pre-defined options.

The State of Agile survey gave five tips for success with scaling agile: consistent process and practices (43%), implementation of a common tool across teams (40%), agile consultants or trainers (40%), executive sponsorship (37%) and internal agile support team (35%). Four of these tips match very well to our findings. The first one of these, the consistent process and practices, is very close to our success factor "confirm to a single approach". The third one, agile consultants or trainers, match to our training and coaching category, where we have as success factors "provide training on agile methods" and "coach teams as they learn by doing". The fourth tip, executive sponsorship, fit to our success factor category "management support". Finally, the last tip, internal agile support team, is at least partially included in our success factor "coach teams as they learn by doing", as in many of the cases that mentioned this, the day-to-day coaching was provided by internal coaches. However, the second tip, implementation of a common tool across teams, did not rise as a success factor in our study,

even though several cases mentioned the implementation of common tools. Nevertheless, they did not lift that up as a success factor for transformation.

5.3. Discrepancies and open issues

The papers included in our review provided several pieces of good advice in the form of challenges and success factors. However, looking at the challenges and success factors, we can notice discrepancies, and even controversial advice, highlighting aspects of agile transformation that would require deeper study to gain more insight.

Dealing with the old process. It is unclear how and when to completely drop the old process and its related concepts and practices. The literature mentions challenges when keeping the old way of working, such as using the "Old and new models side by side", "Management in waterfall mode", and "Integrating non-development functions in the transformation" indicating that the coexistence of two different ways of working is problematic. On the other hand, most transformations were done using a stepwise approach, meaning that this situation will exist at least for some while in the organization, and using the old way of working as a reference for the transformation can also be beneficial, as the good practice "Map to old way of working to ease adaptation" suggests.

Drawing the line between "one single approach" and allowing self-organization. On the one hand, to successfully perform an agile transformation, it seems important to use a single approach as a starting point, as the success factor: "Conform to a single approach" indicates. This allows the organization to have a consistent vocabulary, which was seen to benefit the organization and support the change [P8, P16, P43]. A single approach makes it possible to compare work between teams and makes it easier for people to relocate and have predictable progress [P43]. If such an approach is lacking, challenges like "interpretation of agile differs between teams", becomes exacerbated, making it difficult to collaborate between teams or change team members between the teams. On the other hand, some primary sources suggest that teams should be allowed to self-organize and customize their practices individually to fit the needs of each team, and full control should be given to teams. It was also mentioned that in a larger organization it is not feasible to use the same process for all projects [P49], but individual application of the agile process is needed for different types of development, depending, e.g., on the type of the software being developed or the project size [P45].

Providing management support without suppressing grass-root level empowerment. On the one hand, having management support is a necessary condition for successful transformation, as shown by the related success factors. However, forcing the transformation from the top can create problems, as some organizations reported "Top-down mandate creates resistance" as a challenge. Of the reported transformations, about half were led top-down, and about one forth bottom-up, and one fourth using a hybrid model.

5.4. Limitations

Researcher bias might have influenced the selection of primary studies, as well as data extraction. The selection of the primary studies may have been distorted by interpreting the inclusion criteria falsely. This risk was mitigated by using three researchers in designing the inclusion criteria. When the inclusion criteria was subsequently applied, the abstract filtering was performed independently by two researchers, and unclear cases were resolved by case-by-case discussions between two or three researchers. In the full-text filtering phase, one researcher did the initial filtering making the decision regarding the unambiguous cases. For the less clear-cut cases—over half of the papers—two researchers read the paper and independently made decision proposals regarding inclusion or exclusion. In cases they agreed, the paper was included or excluded based on their joint agreement. Papers that still were unclear were discussed and resolved by all three researchers together.

The second part of the research that may have been affected by subjective bias was the elicitation of results by coding and analysis. Our tools for mitigating this threat were limited, as the work stages in question are particularly laborious. For resource constraints the process could not be duplicated. We made an attempt to prevent subjective bias in analysis by making the results as traceable as possible, by supplying references to each claim presented in the results.

A particular problem in this systematic literature review was the limitations of Boolean keyword searches in online databases. A keyword search cannot easily identify the facets *large-scale* and *empirical*, and also the facet *transformation* is difficult to capture with keywords. For these reasons we did not include the facets *large-scale* and *empirical* in the keyword search, but instead did a manual walkthrough of all the selected papers in the filtering phase to determine the size of the organization and whether the paper contained empirical material. This added manual work, but mitigated the threat.

For the facet *transformation*, we used a variety of synonyms as keywords, but a small risk remains that some studies discussing transformations without using any of the keywords we used remain unidentified.

Through our preliminary search we found three relevant papers that were not found by our keyword search, and that we included in the study as primary sources. In addition, by going through the references of the 170 papers selected for full text filtering we identified two additional relevant papers that had not been spotted by the keyword search and that we included as primary studies. By these means we aimed to make sure to locate papers that the keyword search might have missed. The fact that it was possible to find additional papers outside the search result suggests that a possibility remains that our keyword search missed some other papers as well. However, because we put reasonable effort in attempting to identify missed publications, we believe that the number of possibly missed papers remains very limited, and thus does not significantly affect the results.

Only six of the primary studies presented a clearly defined research method. Most of the papers were experience reports, in most of which author was a member of the organization discussed, as shown in Table 6, creating author bias. However, due to the low number of research papers, we deemed that the results would be distorted heavily and many valuable studies would be left out if a strict quality assessment would be part of the inclusion criteria. As a result we decided to include all experience reports, regardless of the perceived objectivity.

Many primary studies were openly pro-agile, without giving a solid motivation for the standpoint. The tendency to publish only positive results is another particular problem for this literature review. The primary studies typically reported the transformations as successful, a sign of publication bias. A related issue was that the majority of experience reports were authored by persons personally involved in the transformation, due to which the authors might be reluctant towards reporting problems in the transformation. However, most papers did bring up several challenges experienced during the transformation as well as perceived success factors. Due to the current state of research it is necessary to include studies with varying strength of evidence in order to relevantly aggregate evidence on large-scale transformations.

Due to the author and publication bias of the primary studies and qualitative nature of transformation descriptions, we decided not to make quantitative interpretations in the results, instead use only qualitative analysis. Therefore, based on this data

we could not, e.g., rank the challenges and success factors and say which ones are the most important ones. However, by looking at the number of cases that mentioned some success factor or challenge gives an indication about the prevalence of the factors in the existing literature. However, as we do not know how well the published papers represent the industry, we cannot generalize e.g. the percentages to the industry at large.

5.5. Future research agenda

Based on our literature review, we identify five topics that we think requires significant further study, and that can together be considered to form a research agenda on large-scale agile transformations.

1) Case studies on transformations. More research conducted with proper methods on large-scale agile transformations is acutely needed. In our review, we were able to identify only six research papers on the topic, despite the large practitioner interest in the topic. The timeliness and importance to practitioners is evidenced, e.g., through the fact that recent years have seen a number of books published and courses organized by consultants, and a large number of talks on this topic at agile conferences. Thus, the found 46 experience reports and six research papers seem to only scratch the surface and report experiences from only a small fraction of the real number of transformations happening. Thus, we suggest more case studies on understanding the large-scale agile transformations better and how they are done in practice.

2) Scaling practices. In our review, we scoped out the practical adaptations and augmentations, "scaling practices", used by the organizations. Some agile methods, e.g., Scrum have suggestions: Scrum proposes the use of Scrum-of-Scrums as the main scaling practice. Several frameworks by consultants have been developed and are actively promoted, e.g. SAFe and LeSS. These frameworks introduce several additional practices for scaling. E.g., LeSS suggests practices like Area Product Owners and Communities of Practice. Yet, little research has been done on what scaling practices actually are used in companies, or of their related challenges and benefits or context-dependency.

3) Scaling frameworks. Consultants and practitioners have put forward several frameworks for scaling agile. For example, agile consultants have put together an "Agile Scaling Knowledgebase Decision Matrix" (Mat, 2016), where they briefly compare different agile scaling approaches in one big excel sheet. Currently, they have listed nine different approaches. While going through the articles for this review, we noticed that experiences reports and research papers very seldom mention any frameworks. To our knowledge, there exists only a handful of papers on actual usage experiences related to a particular scaling framework. This is in stark contrast to the fact that companies report using these frameworks, e.g., according to Version One's 10th survey on the state of agile 23% of respondents reported using SAFe and 4% LeSS. Thus, it would be important to study the scaling frameworks scientifically: How much are these scaling frameworks used? How are they used? What are the benefits and challenges of using them? To what kind of circumstances is each of them suitable? How much are they tailored in practice to the needs of the customers? As the reported usages of the scaling frameworks in the scientific literature is low, we encourage researchers to perform in-depth case studies to understand how the frameworks are used and how they are possibly customized.

4) Enterprise agile. As one of our challenge groups, "Integrating non-development functions in the transformation", showed the agile adoption often started from the development organization, and other parts of the organization found it difficult to adapt. However, the development organization operates in a larger context and needs to interface with other organizational functions. This is challenging if the other functions do not adapt to or adopt agile, and are unwilling to change. Thus, for ensuring the success of the whole transformation, it seems to be important that other organizational functions support and adopt agile. It would be interesting to study how these other functions can best be included in, and support an agile transformation at the enterprise scale.

5) Surveys on challenges and success factors. Surveys on challenges and success factors for agile projects in general have been conducted, e.g.(Chow and Cao, 2008). However, specifically large-scale agile projects have not been scientifically studied. Non-scientific surveys exist, the most famous one being the State of Agile Survey that Version One has been conducting annually since 2007. That survey has recently asked a few questions related to large scale as well, e.g. on scaling methods used and tips for success with scaling agile. According to the latest survey (VersionOne, Inc, 2016), 62% of the almost 4000 respondents had more than hundred people in their software organization and 43% of all the respondents worked in development organizations where more than half of the teams were agile. Of course, the sample of this study is limited to a selected subset of companies and countries (of the almost 4000 respondents to the latest survey 65% were from North America and 26% from Europe). However, this indicates that there seems to exist a large number of companies that have taken or are taking large-scale agile into use. Thus, there is a possibility to perform interesting survey studies as well. One topic for further surveys would be to study how the challenges and success factors recognized in this study are experienced in the companies: which ones they have experienced and which ones they consider most important.

6. Conclusions

We presented a systematic literature review of empirical studies and experience reports on large-scale agile transformations. We analyzed 52 papers describing 42 different organizations, presenting qualitative findings describing reported challenges and success factors for large-scale agile transformations.

The identified primary studies were almost exclusively experience reports, only six research papers were included. Thus, the main finding of this study is that despite the relevance of the topic for practitioners, research is seriously lagging behind, and as a consequence the identified success factors and challenges are those that practitioners perceive and report as most important. The relationship between these and objective fact remains unknown.

The challenge categories that received the most mentions are *agile difficult to implement, integrating non-development functions, change resistance*, and *requirements engineering challenges*. The success factor categories that received the most mentions are *choosing and customizing the agile approach, management support, mindset and alignment*, and *training and coaching*.

As future research topics we suggest case studies on agile transformations, studies on the usage of scaling practices and scaling frameworks, as well as enterprise wide use of agile, and surveys on large-scale agile.

Acknowledgment

This study was funded by TEKES as part of the Need for Speed (N4S) SHOK program.

References

Anonymous, Agile Software Development and the Factors that Drive Success, Technical Report, Forrester Research Inc., 2012.

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

Berger, H., Beynon-Davies, P., 2009. The utility of rapid application development in large-scale, complex projects. Inf. Syst. J. 19 (6), 549–570.

Bjarnason, E., Wnuk, K., Regnell, B., 2011. A case study on benefits and side-effects of agile practices in large-scale requirements engineering. In: Proceedings of the 1st Workshop on Agile Requirements Engineering. In: AREW '11, pp. 3:1–3:5.

Boehm, B., 2002. Get ready for agile methods, with care. Computer 35 (1), 64–69.

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

Chow, T., Cao, D.-B., 2008. A survey study of critical success factors in agile software projects. J. Syst. Softw. 81 (6), 961–971. doi:10.1016/j.jss.2007.08.020.

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

Cockburn, A., Highsmith, J., 2001. Agile software development, the people factor. Computer 34 (11), 131–133. doi:10.1109/2.963450.

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

Cruzes, D., Dyba, T., 2011. Recommended steps for thematic synthesis in software engineering. In: Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement. In: ESEM '11. IEEE Computer Society, Washington, DC, USA, pp. 275–284.

Dingsøyr, T., Fægri, T., Itkonen, J., 2014. What is large in large-scale? a taxonomy of scale for agile software development. In: Jedlitschka, A., Kuvaja, P., Kuhrmann, M., Männistö, T., Münch, J., Raatikainen, M. (Eds.), Product-Focused Software Process Improvement. In: Lecture Notes in Computer Science, vol. 8892. Springer International Publishing, pp. 273–276. doi:10.1007/978-3-319-13835-0_20.

Dingsøyr, T., Moe, N., 2013. Research challenges in large-scale agile software development. SIGSOFT Softw. Eng. Notes 38 (5), 38–39.

Dingsøyr, T., Moe, N., 2014. Towards principles of large-scale agile development. In: Dingsøyr, T., Moe, N., Tonelli, R., Counsell, S., Gencel, C., Petersen, K. (Eds.), Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation. In: Lecture Notes in Business Information Processing, vol. 199. Springer International Publishing, pp. 1–8. doi:10.1007/978-3-319-14358-3_1.

Dybå, T., Dingsøyr, T., 2008. Empirical studies of agile software development: a systematic review. Inf. Softw. Technol. 50 (9-10), 833–859.

Dybå, T., Dingsøyr, T., 2009. What do we know about agile software development? Softw. IEEE 26 (5), 6–9.

Elshamy, A., Elssamadisy, A., 2006. Divide after you conquer: an agile software development practice for large projects. In: Extreme Programming and Agile Processes in Software Engineering. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4044 LNCS, pp. 164–168.

Federoff, M., Courage, C., 2009. Successful user experience in an agile enterprise environment. In: Procceedings of HCI International 2009, San Diego, CA, USA, July 19-24, Part I. In: LNCS, 5617, pp. 233–242.

Fitzgerald, B., Hartnett, G., Conboy, K., 2006. Customising agile methods to software practices at intel shannon. Eur. J. Inf. Syst. 15 (2), 200–213.

Fowler, M., 2000. Put your process on a diet. Softw. Dev. 8 (12), 32–36.

Fowler, M., Highsmith, J., The agile manifesto (2001). http://www.agilemanifesto.org/.

Freudenberg, S., Sharp, H., 2010. The top 10 burning research questions from practitioners. Softw. IEEE 27 (5), 8–9.

Fry, C., Greene, S., 2007. Large scale agile transformation in an on-demand world. In: Agile Conference (AGILE), 2007, pp. 136–142.

Fulgham, C., Johnson, J., Crandall, M., Jackson, L., Burrows, N., 2011. The FBI gets agile. IT Prof. 13 (5), 57–59.

Giudice, D.L., Kisker, H., Angel, N., 2014. How Can You Scale Your Agile Adoption? Technical Report. Forrester Research Inc.

Hamed, A., Abushama, H., 2013. Popular agile approaches in software development: review and analysis. In: Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on, pp. 160–166. doi:10.1109/ICCEEE.2013.6633925.

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

Hodgkins, P., Hohmann, L., 2007. Agile program management: Lessons learned from the verisign managed security services team. In: Agile Conference (AGILE), 2007, pp. 194–199.

Jalali, S., Wohlin, C., 2012. Global software engineering and agile practices: a systematic review. J. Softw. 24 (6), 643–659.

Kaisti, M., Rantala, V., Mujunen, T., Hyrynsalmi, S., Konnola, K., Makila, T., Lehtonen, T., 2013. Agile methods for embedded systems development - a literature review and a mapping study. EURASIP J. Embedded Syst. 2013 (1), 15.

Kim, E., Ryoo, S., 2012. Agile adoption story from NHN. In: Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual, pp. 476–481.

Kitchenham, B.A., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical report EBSE-2007-01. Keele University.

Koehnemann, H., Coats, M., 2009. Experiences applying agile practices to large systems. In: Agile Conference, 2009. AGILE '09., pp. 295–300.

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), 26–34.

Livermore, J.A., 2008. Factors that significantly impact the implementation of an agile software development methodology. J. Softw. 3 (4), 31–36.

Lyon, R., Evans, M., 2008. Scaling up – pushing scrum out of its comfort zone. In: Agile, 2008. AGILE '08. Conference, pp. 395–400.

Maranzato, R., Neubert, M., Herculano, P., 2012. Scaling scrum step by step: the mega framework. In: Agile Conference (AGILE), 2012, pp. 79–85.

Miller, J., Haddad, H., 2012. Challenges faced while simultaneously implementing cmmi and scrum: a case study in the tax preparation software industry. In: Information Technology: New Generations (ITNG), 2012 Ninth International Conference on, pp. 314–318.

Mishra, D., Mishra, A., 2011. Complex software project development: agile methods adoption. J. Softw. Maint. Evolution 23 (8), 549–564.

Misra, S.C., Kumar, V., Kumar, U., 2010. Identifying some critical changes required in adopting agile practices in traditional software development projects. Int. J. Qual. Reliab. Manag. 27 (4), 451–474.

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

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

Paasivaara, M., Behm, B., Lassenius, C., Hallikainen, M., 2014. Towards rapid releases in large-scale xaas development at ericsson: A case study. In: Global Software Engineering (ICGSE), 2014 IEEE 9th International Conference on, pp. 16–25. doi:10.1109/ICGSE.2014.22.

Paasivaara, M., Durasiewicz, S., Lassenius, C., 2008. Using scrum in a globally distributed project: a case study. Softw. Process Improv. Pract. 13 (6), 527–544.

Paasivaara, M., Lassenius, C., Heikkila, V., Dikert, K., Engblom, C., 2013. Integrating global sites into the lean and agile transformation at ericsson. In: Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on, pp. 134–143. doi:10.1109/ICGSE.2013.25.

Petersen, K., Wohlin, C., 2010. The effect of moving from a plan-driven to an incremental software development approach with agile practices. Emp. Softw. Eng. 15 (6), 654–693.

Sagesser, K., Joseph, B., Grau, R., 2013. Introducing an iterative life-cycle model at credit suisse it switzerland. Softw. IEEE 30 (2), 68–73.

Schwaber, K., Beedle, M., 2002. Agile Software Development with Scrum. Series in Agile Software Development. Prentice Hall.

Scott, J., Johnson, R., McCullough, M., 2008. Executing agile in a structured organization: government. In: Agile, 2008. AGILE '08. Conference, pp. 166–170.

Senapathi, M., Srinivasan, A., 2013. Sustained agile usage: a systematic literature review. In: Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering. In: EASE '13. ACM, pp. 119–124.

Spearman, S. and Dolman, R. Agile scaling knowledgebase decision matrix, 2016. http://www.agilescaling.org/ask-matrix.html. Cited 10.6.2016.

VersionOne, Inc, 10th annual "state of agile development" survey, 2016.

Williams, L., Cockburn, A., 2003. Agile software development: it's about feedback and change. Computer 36 (6), 39–43.

M.Sc. Kim Dikert is a software engineer and software entrepreneur with an interest in software engineering research. As a practitioner he works as a software architect. His research interests include agile software engineering in small and large contexts.

Dr. Maria Paasivaara is a research fellow at Aalto University. Her research interests include global software engineering, scaling agile and lean software engineering, and software engineering education. She has a D.Sc. degree from Aalto University.

Dr. Casper Lassenius is an associate professor at Aalto University. His current research interests include agile and lean software development, global software engineering, and software quality assurance. He has a D.Sc. degree from Aalto University.