



**HAL**  
open science

# Implementing Large-Scale Agile - an Analysis of Challenges and Success Factors

Björn Kischelewski, Janek Richter

► **To cite this version:**

Björn Kischelewski, Janek Richter. Implementing Large-Scale Agile - an Analysis of Challenges and Success Factors. 28th European Conference on Information Systems (ECIS), Jun 2020, Marrakech, Morocco. pp.176. <hal-04603145>

**HAL Id: hal-04603145**

**<https://hal.science/hal-04603145v1>**

Submitted on 11 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# IMPLEMENTING LARGE-SCALE AGILE - AN ANALYSIS OF CHALLENGES AND SUCCESS FACTORS

*Research paper*

Kischelewski, Björn, University of Cologne, Cologne, Germany, bjoern@kischelewski.de

Richter, Janek, University of Cologne, Cologne, Germany, janek.richter@wiso.uni-koeln.de

## Abstract

*Agile methods have become a standard in software development projects and large organizations are applying these methods to their projects. The implementation of agile methods in contexts of large and multiple teams that may span across different locations is referred to as large-scale agile. Limited systematic and scientific knowledge about large-scale agile exists that is based on empirical research. Aggregated knowledge is primarily based on experience reports and industry cases. Hence, organizations face difficulties to successfully implement large-scale agile. We conduct an analysis of challenges and success factors that we identified in empirical research. The majority of challenges and success factors correspond to topics such as communication, coordination, and knowledge management. We discuss our results by highlighting distribution among agile methods (e.g. Scrum) and agile frameworks that are designed for large-scale agile by definition (e.g. SAFe). We conclude our paper by discussing limitations and presenting avenues for future research. Our paper contributes to the body of knowledge on large-scale agile implementations and provides a systematic overview of challenges and success factors.*

*Keywords: Agile Software Development, Large-Scale, Challenges, Success Factors, Literature Analysis.*

## 1 Introduction

Agile methods and practices such as Scrum or Extreme Programming (XP) are software development methods based on the agile manifesto (Beck et al., 2001). The goal of these methods is to reduce the development costs resulting from change (Highsmith and Cockburn, 2001). Therefore, agile methods rely on different values and principles. The core differences to a plan-driven, waterfall approach are focus on customers, embracing change, focus on working software and requiring intense interactions in teams (Beck et al., 2001; Williams and Cockburn, 2003).

Since the publication of the agile manifesto in 2001, agile methods have become a standard in software development projects (Dingsøyr and Moe, 2014). As the latest Global Project Management Survey shows, 73% of the participating companies use agile practices at least sometimes (Project Management Institute, 2018). Further, a study by PwC finds that projects using agile methods are 28% more successful than projects using traditional practices (Jonnalagadda et al., 2017). At the same time, agile software development has gained increasing interest in research (Dingsøyr et al., 2012).

Parallel to the increasing interest in agile software development, large organizations, which are in a process of digital transformation, are applying these agile methods to their projects (Dingsøyr and Moe, 2014). Today, agile methods are also used for large projects with multiple teams which may span across different locations. This implementation of agile methods is referred to as *large-scale agile* (Dingsøyr and Moe, 2014; Dingsøyr et al., 2018b). Though, a common definition of large-scale agile is missing in research, a workshop at the XP2017 conference defines three to five teams involved in the project as a lower bound for when an agile project is large-scale. (Moe and Dingsøyr, 2017).

Agile methods were originally designed to fit small, co-located teams developing non-life-critical projects (Williams and Cockburn, 2003). Therefore, the implementation of large-scale agile challenges several of the core assumptions of agile methods (Dingsøy and Moe, 2014). Since increasing numbers of large organizations have started to implement large-scale agile projects in the last decade, limited systematic and scientific knowledge about implications exists. For example, consensus about the most relevant challenges and success factors which come with agile at scale does not exist yet (Ebert and Paasivaara, 2017). In addition, there is limited knowledge on how different frameworks impact these challenges and success factors (Dikert et al., 2016; Ebert and Paasivaara, 2017). Hence, organizations face difficulties to successfully implement large-scale agile. This knowledge gap is highly relevant for most large organizations because fast changing markets with highly volatile customer demands, which create a high probability of change during a project, have become common (Laanti, 2014). Thus, large organizations started implementing agile methods to reduce their costs of change. Further, it is important for these organizations to find a way to implement agile methods at scale which reduces the risks and thereby the costs of this process. Consequently, we pose the following research question: Which challenges and success factors for large-scale agile implementations exist in empirical research and how are they related with agile frameworks and methods?

Research is lacking comprehensive overviews of what we already know about the implementation of large-scale agile. Due to the lack of availability of peer-reviewed empirical research, recent literature reviews focused on experience reports and industry cases (e.g., Dikert et al., 2016). However, these reports have to be addressed with caution as they may not have gone through peer-review process controlling for quality of scientific method and they are often published by directly involved parties. An up-to-date overview of empirical research enables researchers and large organizations to identify important opportunities and challenges related to implementation of large-scale agile projects. Further, it offers a starting point for future research by revealing directions, gaps and biases in literature.

We conduct a systematic literature review that analyzes the findings of empirical research on the implementation of large-scale agile projects. By analyzing challenges and success factors for large-scale agile implementations, we provide an overview of topical empirical literature and how it evolved. We contribute by offering insights for organizations on how to conduct a large-scale agile implementation and transformation.

## **2 Theoretical Background**

### **2.1 Large-scale Agile**

Agile methods were designed to reduce the high cost – compared to traditional and plan-driven development methods such as Waterfall – resulting from changes during development (Highsmith and Cockburn, 2001). In comparison to traditional development methods, agile methods embrace change and are designed with quick adaptations in mind (Highsmith and Cockburn, 2001; Miller, 2001; Williams and Cockburn, 2003). Since 2001, the agile manifesto serves as a foundation for agile methods (Beck et al., 2001; Williams and Cockburn, 2003). By definition, agile methods best suit “non-safety-critical projects with volatile requirements, built by relatively small and skilled co-located teams” (Williams and Cockburn, 2003, p. 40). Williams and Cockburn (2003) limit the optimal team size for agile methods to 50 members.

Drawing from this sweet spot for team size, agile methods are not well-suited for large-scale projects (Dingsøy and Moe, 2014). Still, modern large organizations increasingly use agile methods for large and distributed projects with composite structures (Dingsøy and Moe, 2013; Dingsøy and Moe, 2014). This implementation of agile methods is referred to as large-scale agile in literature (Dingsøy and Moe, 2014; Dingsøy et al., 2018b). The exact definition of large-scale agile projects differs among authors. Possible metrics to assess large-scale agile projects are costs, lines of code, number of features, number of teams involved and people involved (Dingsøy et al., 2014). As suggested by Dingsøy et al. (2014), a suitable metric should measure the coordination effort. At the workshop on large-scale agile implementations at the XP2017 conference the participants defined three to five teams as minimum for large-

scale agile projects (Moe and Dingsøy, 2017). To ensure a clear delimitation to agile in the small scale we define large-scale agile implementations as all projects following one or several agile methods with at least four teams involved. Further, we define 50 people as a lower bound for the total size (including team members plus supporting roles).

## 2.2 Large-scale Agile Methods and Frameworks

Since the implementation of agile methods in a large scale is becoming increasingly popular, methods and frameworks to structure the process of large-scale agile projects have emerged. The five most used large-scale agile methods are the Scaled Agile Framework (SAFe), Scrum of Scrums, Disciplined Agile Delivery (DAD), the Spotify Model and Large Scale Scrum (LeSS) (VersionOne and CollabNet, 2019). SAFe combines the five layers of “Lean Enterprise-Lean-Agile Leadership, Team and Technical Agility, DevOps and Release on Demand, Business Solutions and Lean Systems Engineering, and Lean Portfolio Management” (Scaled Agile Inc., 2018, p. 3). Each layer consists of its own roles and practices and all layers are linked by the framework. At the team level, SAFe implements Scrum teams which are supported by practices of Kanban and Lean. The overall goal of SAFe is to “enable enterprises to achieve the best quality and value in the shortest sustainable lead time” (Scaled Agile Inc., 2018, pp. 2-3).

Scrum of Scrums handles the coordination of multiple Scrum teams which are working on the same project (Mundra et al., 2013). The basic coordination mechanism used is the Scrum of Scrums meeting in which at least one representative of each team takes part (Paasivaara et al., 2012). It is recommended that this meeting takes place once a day to three times a week (Cohn, 2007; Sutherland et al., 2007). Its purpose is not a pure status update but solving dependency problems and pushing forward the integration (Gustavsson, 2019).

DAD is a hybrid large-scale agile framework which extends Scrum by other agile methods to address a project’s full, end-to-end delivery lifecycle. Its most important characteristics are people first, learning-oriented, agile, hybrid, IT solution focused, goal-driven delivery life cycle, risk and value driven and enterprise aware (Ambler and Lines, 2011).

The Spotify Model is a framework based on the large-scale agile implementation at the company Spotify. It works with several self-organized teams called squads which each have a long-term mission which adds to the product. Each squad is free to choose its agile methods and practices (Alqudah and Razali, 2016).

With LeSS, the whole project has one main product owner who is responsible for the backlog and prioritizes the requirements. At the beginning of each sprint, an overall sprint planning is conducted in which the product owner and two members of each individual team participate and afterwards, each team conducts its own sprint planning. During the sprint, a product backlog refinement meeting is conducted at which all team members of all teams participate and optionally, inter-team coordination meetings take place. At the end of each sprint, a sprint review is conducted at which the product owner, two members of each team and other stakeholders participate and additionally, each individual team does its own retrospective (Larman and Vodde, 2013).

## 3 Method

To answer our research question, we conducted a systematic literature review, in which we focus on empirical research on large-scale agile implementations (Webster and Watson, 2002). The literature review process consists of three steps: literature search, selection, and analysis.

### 3.1 Literature Search and Selection

We depict the literature search and selection process in Figure 1. We implemented a preliminary search to identify relevant keywords and terminology (Rowley and Slack, 2004). The relevant results of the preliminary search were used to build a keyword pool which contains phrases such as *agile practices*, *enterprise-wide*, *scaled*, *at scale*. The final search string was developed upon the building block strategy

(Rowley and Slack, 2004). On the top level, our final search string logically combines terms of the field *agile* and *large-scale* (i.e. ‘*agile AND large-scale*’). On the lower level, terms are logically combined with an OR-operator (e.g. *large-scale OR large enterprise OR multi-team projects*).

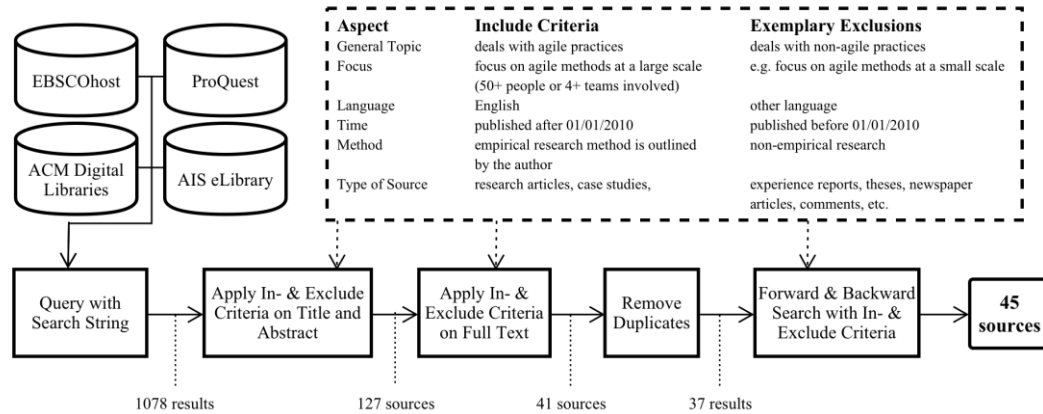


Figure 1. Literature Search Process

As recommended by literature (Kitchenham, 2004), we present our include criteria in Figure 1. The literature has to deal with agile methods implemented in a large-scale context (as defined in Section 2). Further, we focus on articles published after January 1<sup>st</sup> 2010 in alignment with the prominent highlighting of this research topic of ‘agile and large projects’ in the context of XP2010 conference (Freudenberg and Sharp, 2010). English literature was prerequisite as being the established language of scholarly and technological communication. We particularly focus on empirical research in comparison to previous literature reviews on the topic that include experience reports and industry cases to a large degree (e.g. Dikert et al., 2016). We only included articles where authors explicitly state the empirical method applied. The type of sources should be research articles and case studies.

To achieve broad coverage of relevant literature, we used various databases for scientific literature (EBSCOhost’s Academic Search Complete and Business Source Complete, ProQuest, ACM Digital Libraries and AIS eLibrary). Our search yielded a total of 1078 articles. A first screening round was conducted based on title, keyword and abstract, resulting in 127 candidate articles. In the subsequent selection, we considered articles’ full texts and eliminated duplicates – resulting in 37 articles relevant for our review. Next, we conducted a forward and backward search based on the relevant literature (Webster and Watson, 2002). For the backward search, previous literature reviews were considered in order to find relevant papers as recommended by methodological literature (Rowley and Slack, 2004). Forward search was conducted using Web of Science (Webster and Watson, 2002). The research agenda resulting from XP2014 conference was included as a basis for the forward searches as well (Dingsøy and Moe, 2014). Forward and backward search yielded eight additional articles. Thus, we identified 45 core articles in total that are included in the literature analysis process.

### 3.2 Literature Analysis

We inductively coded the articles for challenges and success factors for large-scale agile implementations resulting in a concept matrix (Rowley and Slack, 2004; Webster and Watson, 2002). Articles based on the same data and presenting the same case(s) were merged on their findings of challenges and success factors. Merged articles are highlighted in Table 1 in the appendix. If we identified a challenge or success factor in at least two articles it became a candidate for our concept matrix. The analysis resulted in 18 challenges and 24 success factors. In the next step, we created categories of challenges and success factors (Webster and Watson, 2002). Although we started with a preliminary set of categories from prior literature, we added, merged, and revised these when necessary while assigning all challenges and success factors. This process resulted in six categories of challenges and seven categories of success factors.

Agile methods included in our core articles were coded as well. An article was allocated to a specific agile method if at least one of its analyzed cases applied this method. We categorize agile methods into large-scale agile frameworks and traditional agile methods to discuss the impact of large-scale agile frameworks.

## 4 Results

### 4.1 Overview of Selected Articles

The final literature selection includes a total of 45 articles listed in Table 1 in the appendix. All these articles empirically examine large-scale agile by analyzing concrete cases. The number of publications based on the year of publication is depicted in Figure 2. The year 2019 is excluded from the figure since we conducted the literature search in June 2019 and identified five articles in the current year. The low number of papers published in 2010 and 2011 reflects the fact that ‘agile and large projects’ was ranked first among the most relevant research issues that need to be addressed in the future at the XP2010 (Freudenberg and Sharp, 2010). A further increase of publications beginning with the year 2016 can be explained by the research agendas which have been published at the XP2013 and XP2014 conference (Dingsøy and Moe, 2013; Dingsøy and Moe, 2014).

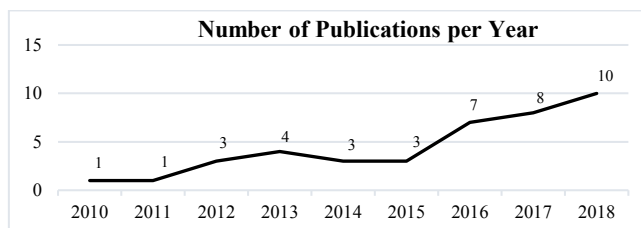


Figure 2. Temporal Distribution of Articles

Our core set of 45 articles empirically analyze real-world implementations of agile projects in a large-scale. The majority of author teams analyze one case (27 articles). At maximum, one article comprises ten cases (Gerster et al., 2018b). Concerning data collection, authors use six different methods (i.e. interviews, surveys, analysis of documents, observations, focus groups) and 22 articles entail more than one data method. Authors of nearly all articles used interviews as data collection method (41 articles) and are based on primary data (43 articles).

We identified a total of nine different agile methods used within our analyzed set of articles. Agile methods can be distinguished between traditional agile methods, which are designed for small projects, and large-scale agile methods and frameworks, which are particularly developed for large-scale agile implementations. These large-scale agile frameworks are, for example, Scrum of Scrums, SAFe, LeSS, DAD or the Spotify Model. Regardless of whether articles contain traditional agile methods or large-scale agile frameworks, all projects are implemented in a large-scale context. For, example a study reports the usage of XP or Scrum by each team in a large-scale agile project (Heijden et al., 2018). The distribution of used (large-scale) agile methods and frameworks are depicted in Figure 3 and Figure 4 (an article may correspond to several methods and frameworks).

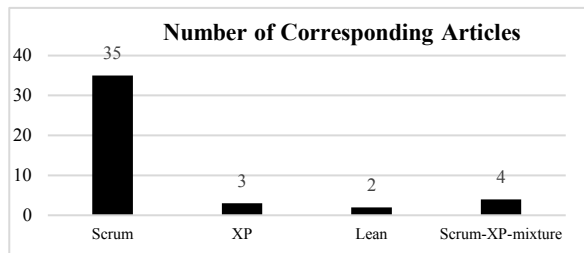


Figure 3. Distribution of Traditional Agile Methods

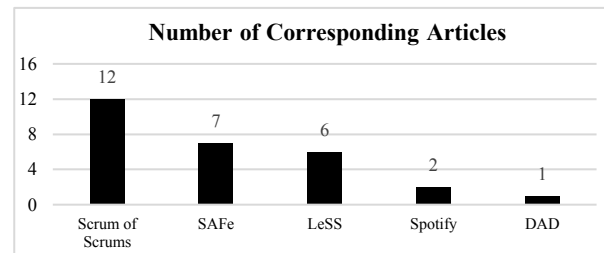


Figure 4. Distribution of Large-scale Agile Frameworks

## 4.2 Challenges for Large-scale Agile

We identify 18 different challenges for large-scale agile implementations. We group challenges into six categories (see Figure 5). Each of the following sub-sections relates to one category.

| CHALLENGES                                |   |           |  |           |
|---|---|-----------|--|-----------|
|   | * | †         |  |           |
| <b>Communication Challenges</b>           |   | <b>16</b> | <b>HR Management Challenges</b>            | <b>11</b> |
| Inter-team Meetings not Working           | 7 |           | Conveying of Agile Mindset                 | 7         |
| Missing Inter-team Communication          | 6 |           | Need of new Positions                      | 3         |
| Schedule Meetings                         | 4 |           | Ensuring Team Competences                  | 3         |
| Temporal Distance                         | 4 |           | Transformation of Old to New Positions     | 2         |
| Barriers to Knowledge Exchange            | 3 |           | <b>Complex Technology</b>                  | <b>8</b>  |
| <b>Inter-team Coordination Challenges</b> |   | <b>12</b> | Integration with other Systems             | 6         |
| Coexistence of Agile and Non-agile Units  | 7 |           | IT Landscape Complexity                    | 4         |
| Resource Conflicts between Teams          | 6 |           | <b>Requirements Engineering Challenges</b> | <b>7</b>  |
| Interfacing of different Agile Practices  | 2 |           | Effort Overrun                             | 4         |
| <b>Culture</b>                            |   | <b>6</b>  | Weak Prioritization                        | 3         |
| Unfitting Company Culture                 | 6 |           | Long-time Planning                         | 2         |

\* number of articles that address the respective challenge

† number of articles that address at least one of the respective category's challenges

Figure 5. Overview of Challenges for Large-scale Agile Projects

### 4.2.1 Communication Challenges

In terms of communication challenges, most articles address the challenge *inter-team meetings not working*. We find that the setup of fruitful Scrum of Scrums meetings itself is a major challenge in large-scale agile. For example, teams assume “that the other teams do not have to know or are not interested in their problems” (Paasivaara et al., 2012, p. 237). In addition, participants are often unable to understand all of the other teams’ problems (e.g. Nyrud and Stray, 2017). The second most identified challenge is *missing inter-team communication*. Communication gaps arise especially between teams of different products (e.g. Uludağ et al., 2019). These can for example lead to “merely local optimizations at the expense of the overall system” (Rodríguez et al., 2013, p. 8). In four articles we identify *temporal distance* as a challenge resulting in missing or bad communication across employees that work in different time zones around the globe. Due to the resulting temporal distance, communication challenges arise between team members and management roles, as for example the product owners, as direct consultation is only feasible during a short time slot (Razzak et al., 2018). The challenge of *scheduling meetings* is reinforced by temporal distance (Razzak et al., 2018; Turetken et al., 2017). However, a study also finds challenges concerning the scheduling of meetings in a non-distributed context (Uludağ

et al., 2019). The least articles are concerned with *barriers to knowledge exchange*. Limited contact with domain experts and other teams' knowledge resources, for instance because of these being perceived as unavailable, can result in a such barriers (e.g., Sablis and Smite, 2016).

#### 4.2.2 Human Resource Management Challenges

When it comes to human resource management challenges, the most articles address the challenge of *conveying of agile mindset*. The difficulty is to convince the employees to engage in the new way of working as well as to get involved with continuous improvement of the process (e.g. Pries-Heje and Krohn, 2017). For example, one research describes that some employees were not willing to participate in the agile methods but rather focused on coding only (Gerster et al., 2018a). Further, three articles find the *need for new positions* poses a challenge. Large-scale agile comes with the need for new and different job roles, for example a product owner, a team builder and coach role or a business process analyst, which have to be adequately filled (Gerster et al., 2018a; Hobbs and Petit, 2017; Pries-Heje and Krohn, 2017). On top of this, it exists the challenge of *transformation of old to new positions*. A main reason for this transformation to be challenging is the fear of former middle management employees to lose power within the restructuring (Pries-Heje and Krohn, 2017). In addition, three articles address the challenge of *ensuring team competences*. For example, an article finds that less mature teams, which are teams with little domain knowledge, competencies and little product experience, perform a worse effort estimation and tend to an effort overrun during a large-scale agile project (Usman et al., 2018).

#### 4.2.3 Inter-team Coordination Challenges

Regarding the coordination among teams, the most often addressed challenge is the *coexistence of agile and non-agile units*. The coordination and cooperation of agile and non-agile teams leads to “conflicts due to different objectives, steering and incentives” (Gerster et al., 2018a, p. 7). This challenge especially occurs whilst coordinating product development teams, working with agile methods, and other functionalities, as for example marketing or sales teams, which follow a non-agile approach (e.g. Hobbs and Petit, 2017). But, also among agile teams it exists the challenge of *interfacing of different agile practices*. For example, the mixture of agile methods can pose a challenge when team members change teams and thus, are confronted with new tools and practices (Tendedez et al., 2018). The second most often addressed challenge of this category is *resource conflicts between teams*. Here, especially the arrangement of a limited number of experts creates conflicts (e.g. Gerster et al., 2018a).

#### 4.2.4 Complex Technology Conditions

The most often addressed challenge in the group of complex technology conditions is *IT landscape complexity*. Large projects in general consist of a larger code base which makes it harder for the employees to comprehend the complete system (Gundelsby, 2018). This process is further intensified by the reduction of documentation of the software architecture which comes with agile methods (e.g. Alsaqaf et al., 2019). The second most often addressed challenge is *integration with other systems*. An article states that a system developed using large-scale agile is on average integrated with ten other systems (Hobbs and Petit, 2017). A higher interdependence results in higher organization, integration and testing efforts which for example, can diminish the amount of release dates per year (Hobbs and Petit, 2017; Rolland et al., 2016). This is in conflict with the first agile principle which states “early and continuous delivery of valuable software” (Beck et al., 2001).

#### 4.2.5 Requirements Engineering Challenges

Also, large-scale agile creates challenges for the requirements engineering and planning. The most often addressed challenge is an *effort overrun* by the teams which is the tendency to take too many features into scope and to underestimate the effort. On the one hand, this effect can arise from the business parts which require too many features, on the other hand, it can arise from the development team underestimating the features' workload (Bjarnason et al., 2011). Two possible reasons for the wrong estimations are too less time and too little information used for the requirements engineering (Razzak et al., 2018).

The second most often addressed challenge is *weak prioritization*. The majority of the features is highly prioritized which impedes a useful ordering of the features (e.g. Bjarnason et al., 2011). A study states that this challenge can arise due to the business units' missing familiarity with agile requirements engineering (van Waardenburg and van Vliet, 2013). The least often addressed challenge is *long-time planning*. Large organizations usually plan within a range of five to ten years which is contradictory to the short time planning cycles which are a core principle of agile methods (Kasauli et al., 2017). For example, these can impede an organization's long time budget planning (Gerster et al., 2018a).

#### 4.2.6 Culture

Six articles also address the challenge of an *unfitting company culture*. This challenge is relevant because an unfitting company culture has a negative impact on the whole large-scale agile transformation. Especially existing hierarchical organizational structures with little decision-making authority at the lower level are in conflict with the agile methods and thus create a challenge for large-scale agile (e.g. Hobbs and Petit, 2017).

### 4.3 Success Factors for Large-scale Agile

We identify 24 different success factors for large-scale agile implementations. We group success factors into seven categories (see Figure 6). Each of the following sub-sections relates to one category.

| SUCCESS FACTORS                         |   |           |                                   |   |           |
|---|---|-----------|-----------------------------------|---|-----------|
|   | * | †         |                                   | * | †         |
| <b>Communication &amp; Transparency</b> |   | <b>20</b> | <b>Knowledge Management</b>       |   | <b>15</b> |
| Transparency                            | 7 |           | Domain Knowledge Sharing          | 5 |           |
| Demos                                   | 7 |           | Experts Roles                     | 4 |           |
| Standardized Tools                      | 4 |           | Artefacts & Documentation         | 4 |           |
| Informal, Ad-hoc Conversations          | 4 |           | Forums / Communities of Practice  | 4 |           |
| Informing & Engaging People             | 4 |           | Wikis                             | 3 |           |
| <b>IT Architecture</b>                  |   | <b>9</b>  | <b>Implementation Strategy</b>    |   | <b>7</b>  |
| Architecture Standards                  | 8 |           | Fostering Continuous Improvements | 5 |           |
| Architecture Role                       | 4 |           | Management Support                | 4 |           |
| Reference Architecture                  | 3 |           | Top-down Implementation           | 2 |           |
| <b>Planning</b>                         |   | <b>9</b>  | Change Agents                     | 2 |           |
| Sprint Zero                             | 5 |           | <b>Team Capabilities</b>          |   | <b>6</b>  |
| Continuous Scope & Release-Planning     | 4 |           | Team Maturity                     | 2 |           |
| <b>Training &amp; Coaching</b>          |   | <b>7</b>  | Rotation of Team Members          | 2 |           |
| Trainings                               | 5 |           | Team Building Interactions        | 2 |           |
| Agile Coaches                           | 4 |           |                                   |   |           |

\* number of articles that address the respective success factor

† number of articles that address at least one of the respective category's success factors

Figure 6. Overview of Success Factors for Large-scale Agile Projects

#### 4.3.1 Communication and Transparency

The most often addressed success factors of this category is *transparency* in the whole large-scale agile transformation, as this is found to minimize the impact of its challenges (e.g. Uludağ et al., 2019). For example, transparent communication is important to align the different stakeholders (e.g. Gren et al., 2017). The second most often addressed success factor is the use of *demos*. These help to support the communication and coordination across business and development units as well as between the development teams (e.g. Bjørnson et al., 2018; Tenedez et al., 2018). A further success factor addressed by

four articles is *standardized tools*. Research finds that it is critical to standardize tools across all partners involved in the project (e.g. Gerster et al., 2018a). To not eliminate the flexibility which comes with agile methods, the standardized tools themselves should value flexibility (Tenedez et al., 2018). Four articles address *informing and engaging people* as success factor of this category. Transparent communication of the transformation to large-scale agile is an important tool to engage people and avoid change resistance (e.g. Paasivaara, 2017). Thereby, it is important to communicate the vision of the transformation timely and transparently in order to fight doubts and engage everyone (e.g. Daneva et al., 2013). As form of communication, four articles highlight *informal, ad-hoc communication* as success factor. This is in line with the agile manifesto which states that “[t]he most efficient and effective method of conveying information [...] is face-to-face conversation” (Beck et al., 2001). This form of communication tackles the challenges that arise in formal meetings used in large-scale agile which are described in chapter 4.2.1 (e.g. Faegri and Moe, 2015).

#### 4.3.2 Knowledge Management

Regarding knowledge management, the most often addressed success factor is *domain knowledge sharing*. The organization should provide access to functional knowledge of the business units to the agile teams (e.g. Hannay and Benestad, 2010). Therefore, one case installed a domain owner role which was responsible for the domain knowledge sharing between developers and business units (Daneva et al., 2013). The knowledge sharing can also be supported by *expert roles*. These are installed outside the teams and temporarily assist them by sharing their expert knowledge (e.g. Hobbs and Petit, 2017). Further, four authors address the success factor of *artefacts and documentation*. Though, this factor is not in line with the agile manifesto which values “[w]orking software over comprehensive documentation” (Beck et al., 2001). For example, research finds that large-scale agile needs more product, release and program governance artefacts and documentation compared to agile in the small scale (e.g. Bass, 2016). One kind of supporting documentation is a *wiki* which was also addressed as success factor by three articles. The wiki should contain “[a]ll process description documents, guidelines, and checklists” (Dingsøy et al., 2018c, p. 72). Four articles also address *forums or communities of practice* as success factor. These forums help the participants to share and broaden their knowledge (e.g. Paasivaara and Lassenius, 2014). Further, they show who knows what and thereby make experts visible (Moe et al., 2014).

#### 4.3.3 IT Architecture

When it comes to a large-scale agile project’s IT architecture, the most often addressed success factor is *architecture standards*. In large-scale agile, architecture standards tackle the challenges of the category IT landscape complexity which were described in chapter 4.2.4 (e.g. Bass and Haxby, 2019). Architecture standards and guidelines should be defined at the beginning of the project and have to be properly documented (e.g. Martini et al., 2016). Another success factor is *reference architecture*, which was addressed the least. A reference architecture “is [...] used to disseminate the approved architecture to stakeholders in a development programme” (Bass, 2016, p. 8). Further, research suggests to avoid losing the whole team autonomy in the architecture design, that teams should be able to create their own sub-architectures (e.g. Dingsøy et al., 2018a). Also, four articles address an *architecture role* as a success factor. This expert role provides architectural coordination across the teams and ensures the adherence to the given architecture standards (e.g. Bass and Haxby, 2019). Further, this role can negotiate and balance the architecture requirements of the business and the development units (e.g. Bick et al., 2016).

#### 4.3.4 Implementation Strategy

Also, the implementation strategy of large-scale agile comes up with three success factors. Most often addressed is the factor of *fostering continuous improvements* which is important during and after the process of implementing large-scale agile. These continuous improvements are based on internal feedback which comes from experiments and errors and thus, these should be welcomed (e.g. Pries-Heje and

Krohn, 2017). As structure of this process, *top-down implementation* is addressed as success factor by two articles. This means that the process should be initiated by the management and implemented top-down (Hobbs and Petit, 2017; Vlaanderen et al., 2012). Also, during the process an important success factor is *management support* as this success factor is the second most often addressed factor in this category. Top and middle management should continuously promote and assist the process of implementation. In addition, these roles must be able to redefine their own way of working (Paasivaara and Lassenius, 2019). A further success factor for the implementation addressed by two articles is the collaboration with *change agents* which support the implementation of large-scale agile. On the one hand, a study finds that these agent roles can be filled by employees of the top or middle management who consequently are particularly responsible to support and guide the change (Paasivaara, 2017). On the other hand, a study investigates a case using change agents which are external consultants (Pries-Heje and Krohn, 2017).

#### 4.3.5 Planning

The most addressed success factor of the category planning is the use of a *sprint zero*, which is a phase of a project before the start of the actual development. Although research finds that the activities included in sprint zero differ among cases, some activities are recurring (Hobbs and Petit, 2017). These activities are the collection of the requirements, the planning of the project's sprints, the definition of high-level architecture and the set-up of the product backlog including initial user stories and epics (Alsaqaf et al., 2019; Hobbs and Petit, 2017; Moe et al., 2014). The second most addressed success factor is *continuous scope and release planning*. This practice manages the inflow of requirements and constantly re-prioritizes these (e.g. Bjarnason et al., 2011). Thereby, it tackles the challenges of the category requirements engineering challenges which were described in chapter 4.2.5 (e.g. Bjarnason et al., 2011).

#### 4.3.6 Training and Coaching

Five articles find that *trainings* are a success factor for large-scale agile. Trainings teach the participants an understanding of agile methods and practices and support the adaption of these methods (e.g. Paasivaara, 2017). They are used for all management and team roles and also for external partners (Martini et al., 2013; Paasivaara, 2017). Another success factor is the cooperation with *agile coaches*. External agile coaches are used to conduct and support the agile trainings (e.g. Uludağ et al., 2019). In addition to external coaches, there are also internal coaches installed which continuously support the implementation of large-scale agile frameworks (e.g. Hobbs and Petit, 2017).

#### 4.3.7 Team Capabilities

Regarding the team capabilities category, three success factors are addressed by two articles each. The first success factor is *team maturity* which is found to have a direct link to a team's performance in large-scale agile. A more mature team is also a more agile team with less effort overruns (Gren et al., 2017; Usman et al., 2018). The second success factor is *rotation of team members*, which is used as a practice to enhance the teams' capabilities by improving inter-team collaboration and knowledge sharing (e.g. Dingsøy et al., 2018c). A downside of this practice is the increasing resistance to team member rotations as the team feeling becomes stronger (Dingsøy et al., 2018c). At the same time, *team building interactions* is addressed as a success factor to improve the team feeling and thus, improve the team capability. Activities used in the cases were for example fun events, day outs or shared lunches (Bjørnson et al., 2018; Sundararajan et al., 2014).

## 5 Discussion

Considering our overall results, the challenges *coexistence of agile and non-agile units*, *inter-team meetings not working*, and *conveying of agile mindset* are identified in seven different articles and represent the most frequently identified challenges. *Coexistence of agile and non-agile units* may be especially

demanding during the transformation to large-scale agile as agile and non-agile teams closely coexist at this stage. Furthermore, this challenge is particularly relevant in situations where end-to-end agile – meaning that every function in the organization uses agile methods (Hobbs and Petit, 2017) – is not desirable. In any of these situations, thus, management and definition of interfaces between agile and traditional parts of the organization is necessary. The challenge *inter-team meetings not working* most notably relates to Scrum of Scrums meetings. The recent Annual State of Agile Report finds that Scrum of Scrums is with 16% the second most frequently used large-scale agile method (VersionOne and CollabNet, 2019). On the one hand, this may indicate that the high number of mentions in our literature review results from the high use of this method. On the other hand, the high coverage of this challenge could reflect that the Scrum of Scrums method’s design is too simple to solve the complex challenges of inter-team communication and coordination at large-scale agile. The other frameworks, such as SAFe, LeSS or DAD, are more complex and we identified less occurrences for this challenge in our results. The third major challenge, which is *conveying of agile mindset*, corroborates as well the Annual State of Agile Report that highlights the challenges of “[...] odds with agile values” and “[...] resistance to change” (VersionOne and CollabNet, 2019, p. 12). This finding indicates that a major challenge for large-scale agile transformation corresponds to cultural aspects of the workforce.

The most frequently identified success factor is *architecture standards*, as this success factor is identified in eight different articles. This finding is especially notable since it directly contrasts the agile manifesto stating that “[t]he best architectures [...] emerge from self-organizing teams” (Beck et al., 2001). Thus, this success factor weakens one of the agile principles and underscores that large-scale agile is not entirely coherent with the agile manifesto.

Considering the difference between articles that investigated cases with large-scale agile frameworks (e.g. SAFe) versus those investigating cases only with classical agile methods (e.g. Scrum), we observed some interesting deviances. For example, we only identified the challenges *scheduling meetings*, *inter-team meetings not working* and *transformation of old to new positions* and the success factors *agile coaches*, *change agents*, *architecture role*, and *wikis* in articles that investigated cases with large-scale agile frameworks. This may reflect that cases where scaled frameworks are in place are much more challenged by coordination and documentation issues and thus require roles that support coordination in larger teams such as change agents and architecture experts (Mundra et al., 2013) as well as tools that focus on team documentation such as wikis.

In contrast, we did not identify the challenges *long-time planning* and *weak prioritization* in articles that investigated cases with large-scale agile frameworks. This could be explained by the fact that these frameworks setup a long-term focus. For example, SAFe “includes Agile approaches to estimating, forecasting, and [...] longer-term roadmapping” (Scaled Agile Inc., 2018, p. 21). Furthermore, LeSS supports requirements prioritization by installing a common product owner for the project.

## 5.1 Limitations

One limitation is the newness of the topic reflected by the reasonable number of authors conducting research on the topic. For example, a single author co-authored in six of the 45 articles presented here. The amount of literature that could be included in our review is thus limited. Another limitation is the missing clear delimitation of large-scale agile and distributed software development. Some cases of large-scale agile come along with a distributed setting as well. Thus, challenges and success factors identified may result from the distributed setting but not from large-scale agile methods alone. The clear delimitation is hard to achieve in research and thus, poses a limitation to this literature review. For future research, we recommend controlling for (non-)distributed in large-scale agile settings. Further, our results do not indicate if challenges and success factors solely originate from the large-scale agile setting but can overlap with challenges and success factors from general project management and small-scale agile settings. Finally, we could only analyze the information reported by the authors. For example, some cases may use a large-scale agile framework or method, however, its application is not explicitly stated in the authors’ analysis.

## **5.2 Future Research**

By giving an up-to-date overview of the literature on the implementation of large-scale agile, we present three possible areas for future research. First, the group of communication challenges is identified as the most frequently addressed in large-scale agile. Although success factors partially map to reported challenges, open problems remain. For example, an effective and fruitful setup of Scrum of Scrums meetings still has to be found when scaling. Second, another large part of our results addresses the organization's IT architecture. Though, there are success factors and practices identified in literature to prepare the IT architecture for large-scale agile, concrete architecture design patterns are missing (e.g. Gerster et al., 2018b). A rare example in our study for an architecture design supporting large-scale agile is an API-centric architecture that connects components of a software system using interfaces (Gundelsby, 2018). At this point, additional research is needed to support this success factor or offer alternative solutions. Third, as large-scale agile becomes increasingly common, future research is needed on end-to-end agile and interfacing agile methods (Hobbs and Petit, 2017). The challenge of interfacing teams using different agile methods will increase with a broader adoption of large-scale agile. In this study, we focused on the implementation of large-scale agile in software development. Other functions, such as marketing or sales, are usually left out. Research should also focus on how to integrate these functions in large-scale agile and thereby on how to create end-to-end agile or actively manage interfaces between functions.

## **6 Conclusion**

We set out to answer the question which challenges and success factors for large-scale agile implementations exist in empirical research and how are they related with agile frameworks and methods. We provide a systematic overview of challenges and success factors for large-scale agile implementations by analyzing empirical literature. For organizations implementing large-scale agile, our results highlight critical areas to focus on such as communication, coordination, and knowledge management. Our results indicate that concrete challenges and success factors may depend on the underlying agile method or framework. Frameworks that are designed for large-scale contexts by definition seem to meet requirements for effective planning better than traditional agile methods such as Scrum. Though, these come with inter-team coordination challenges and the need for more detailed architecture standards and more comprehensive documentation. For researchers this may reflect a fruitful area for further research. Our paper contributes to the body of knowledge on large-scale agile implementations.

## 7 Appendix

| Article                    | *  | **      | Article                              | * | **   |
|----------------------------|----|---------|--------------------------------------|---|------|
| Alsaqaf et al. (2019)      | 6  | I       | Moe et al. (2014)                    | 1 | I, S |
| Bass (2016)                | 9  | I, D    | Nyrud and Stray (2017)               | 1 | I, O |
| Bass and Haxby (2019)      | -  | I       | Paasivaara (2017)                    | 1 | I    |
| Bick et al. (2016)         | 5  | I       | Paasivaara and Lassenius (2019)      | 1 | I    |
| Bjarnason et al. (2012) †  | 1  | I, S    | Paasivaara and Lassenius (2016)      | 1 | I    |
| Bjarnason et al. (2011) †  | 1  | I, S    | Paasivaara and Lassenius (2014)      | 1 | I    |
| Bjørnson et al. (2018)     | 1  | I, D    | Paasivaara et al. (2012)             | 2 | I    |
| Daneva et al. (2013)       | 3  | I       | Pries-Heje and Krohn (2017)          | 1 | I    |
| Dingsøy et al. (2018c) †   | 1  | I, D    | Razzak et al. (2018)                 | 1 | S    |
| Dingsøy et al. (2018a) †   | 1  | I, D    | Rodríguez et al. (2013)              | 1 | S, F |
| Eklund and Berger (2017)   | 6  | I, S    | Rolland (2016) †                     | 1 | I, D |
| Faegri and Moe (2015)      | 1  | D, F    | Rolland et al. (2016) †              | 1 | I, D |
| Gerster et al. (2018b) †   | 10 | I       | Sablis and Smite (2016)              | 2 | I, S |
| Gerster et al. (2018a) †   | 10 | I       | Sundararajan et al. (2014)           | 1 | I    |
| Gren et al. (2017)         | 8  | I, S    | Tendedez et al. (2018)               | 1 | I, O |
| Gundelsby (2018)           | 1  | I       | Turetken et al. (2017)               | 1 | I    |
| Gustavsson (2019)          | 3  | I, O    | Uludağ et al. (2019)                 | 1 | I    |
| Hannay and Benestad (2010) | 1  | I       | Usman et al. (2018)                  | 1 | I, D |
| Heikkilä et al. (2015)     | 2  | I, S, O | Heijden et al. (2018)                | 1 | I    |
| Hobbs and Petit (2017)     | 6  | I, S, D | van Waardenburg and van Vliet (2013) | 2 | I    |
| Kasauli et al. (2017)      | 1  | I       | Vlaanderen et al. (2012)             | 4 | I, D |
| Martini et al. (2016)      | 3  | I, S    | Wale-Kolade (2015)                   | 1 | I    |
| Martini et al. (2013)      | 3  | S       |                                      |   |      |

\* number of analyzed cases

\*\* data collection methods: I=interviews; S=survey; D=analysis of documents; O=observations; F=focus groups

† articles which share data sets and were merged with a second article for analysis (identical first author indicates article pairs)

Table 1. Overview of Analyzed Articles

## References

- Alqudah, M. and Razali, R. (2016). "A review of scaling agile methods in large software development." *International Journal on Advanced Science, Engineering and Information Technology* 6 (6), 828-837.
- Alsaqaf, W., Daneva, M. and Wieringa, R. (2019). "Quality requirements challenges in the context of large-scale distributed agile: An empirical study." *Information & Software Technology* 110, 39-55.
- Ambler, S. and Lines, M. (2011). "Disciplined agile delivery: an introduction." *IBM Software, Somers, NY*, 1-20.
- Bass, J. M. (2016). "Artefacts and agile method tailoring in large-scale offshore software development programmes." *Information & Software Technology* 75, 1-16.
- Bass, J. M. and Haxby, A. (2019). "Tailoring Product Ownership in Large-Scale Agile Projects Managing Scale, Distance, and Governance." *IEEE Software* 36 (2), 58-63.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D. (2001). *Manifesto for Agile Software Development*. URL: <http://agilemanifesto.org> (visited on 4th June).

- Bick, S., Scheerer, A. and Spohrer, K. (2016). "Inter-Team Coordination in Large Agile Software Development Settings: Five Ways of Practicing Agile at Scale." In: *Proceedings of the Scientific Workshop Proceedings of XP2016*. Edinburgh, Scotland, UK, pp. 1-5.
- Bjarnason, E., Wnuk, K. and 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*. Lancaster, United Kingdom, pp. 1-5.
- Bjarnason, E., Wnuk, K. and Regnell, B. (2012). "Are you biting off more than you can chew? A case study on causes and effects of overscoping in large-scale software engineering." *Information & Software Technology* 54 (10), 1107-1124.
- Bjørnson, F. O., Wijnmaalen, J., Stettina, C. J. and Dingsøy, T. (2018). "Inter-team coordination in large-scale agile development: A case study of three enabling mechanisms". International Conference on Agile Software Development, 2018. Springer, Cham, 216-231.
- Cohn, M. (2007). *Advice on conducting the scrum of scrums meeting*. Mountain Goat Software Web site. URL: <http://www.mountaingoatsoftware.com/articles/advice-on-conducting-the-scrum-of-scrums-meeting/comments>.
- Daneva, M., Van Der Veen, E., Amrit, C., Ghaisas, S., Sikkil, K., Kumar, R., Ajmeri, N., Ramteerthkar, U. and Wieringa, R. (2013). "Agile requirements prioritization in large-scale outsourced system projects: An empirical study." *Journal of Systems & Software* 86 (5), 1333-1353.
- Dikert, K., Paasivaara, M. and Lassenius, C. (2016). "Challenges and success factors for large-scale agile transformations: A systematic literature review." *Journal of Systems & Software* 119, 87-108.
- Dingsøy, T., Fægri, T. E. and Itkonen, J. (2014). "What is large in large-scale? A taxonomy of scale for agile software development". International Conference on Product-Focused Software Process Improvement, 2014. Springer, 273-276.
- Dingsøy, T. and Moe, N. (2013). Research Challenges in Large-Scale Agile Software Development.
- Dingsøy, T. and Moe, N. B. (2014). "Towards Principles of Large-Scale Agile Development A Summary of the Workshop at XP2014 and a Revised Research Agenda." In: *15th International Conference on Agile Software Development (XP), MAY 26-30, 2014, Rome, ITALY*. pp. 1-8.
- Dingsøy, T., Moe, N. B., Fægri, T. E. and Seim, E. A. (2018a). "Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation." *Empirical Software Engineering* 23 (1), 490-520.
- Dingsøy, T., Moe, N. B. and Ohlsson, H. H. (2018b). "Towards an understanding of scaling frameworks and business agility: a summary of the 6th international workshop at XP2018." In: *Proceedings of the 19th International Conference on Agile Software Development: Companion*. Porto, Portugal, pp. 1-4.
- Dingsøy, T., Moe, N. B. and Seim, E. A. (2018c). "Coordinating Knowledge Work in Multiteam Programs: Findings From a Large-Scale Agile Development Program." *Project Management Journal* 49 (6), 64-77.
- Dingsøy, T., Nerur, S., Balijepally, V. and Moe, N. B. (2012). "A decade of agile methodologies: Towards explaining agile software development." *Journal of Systems & Software* 85 (6), 1213-1221.
- Ebert, C. and Paasivaara, M. (2017). "Scaling Agile." *IEEE Software* 34 (6), 98-103.
- Eklund, U. and Berger, C. (2017). "Scaling agile development in mechatronic organizations: a comparative case study." In: *Proceedings of the 39th International Conference on Software Engineering: Software Engineering in Practice Track*. Buenos Aires, Argentina, pp. 173-182.
- Fægri, T. E. and Moe, N. B. (2015). "Re-conceptualizing requirements engineering: findings from a large-scale, agile project." In: *Scientific Workshop Proceedings of the XP2015*. Helsinki, Finland, pp. 1-5.
- Freudenberg, S. and Sharp, H. (2010). *The Top 10 Burning Research Questions from Practitioners*.
- Gerster, D., Dremel, C. and Kelker, P. (2018a). "'Agile meets non-agile': Implications of adopting agile practices at enterprises." In: *24th Americas Conference on Information Systems*. pp. 1-10.
- Gerster, D., Dremel, C. and Prashant, K. (2018b). "Scaling Agility: How enterprises adopt agile forms of organizational design." In: *39th International Conference on Information Systems*. pp. 1-9.

- Gren, L., Torkar, R. and Feldt, R. (2017). "Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies." *Journal of Systems and Software* 124, 104-119.
- Gundelsby, J. H. (2018). "Enabling autonomous teams in large-scale agile through architectural principles." In: *Proceedings of the 19th International Conference on Agile Software Development: Companion*. Porto, Portugal, pp. 1-4.
- Gustavsson, T. (2019). "Dynamics of Inter-team Coordination Routines in Large-scale Agile Software Development". 27th European Conference on Information Systems, 2019. 1-15.
- Hannay, J. E. and Benestad, H. C. (2010). "Perceived productivity threats in large agile development projects." In: *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. Bolzano-Bozen, Italy, pp. 1-10.
- Heijden, A. V. D., Broasca, C. and Serebrenik, A. (2018). "An empirical perspective on security challenges in large-scale agile software development." In: *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. Oulu, Finland, pp. 1-4.
- Heikkilä, V. T., Paasivaara, M., Rautiainen, K., Lassenius, C., Toivola, T. and Järvinen, J. (2015). "Operational release planning in large-scale Scrum with multiple stakeholders – A longitudinal case study at F-Secure Corporation." *Information & Software Technology* 57, 116-140.
- Highsmith, J. and Cockburn, A. (2001). "Agile software development: the business of innovation." *Computer* 34 (9), 120-127.
- Hobbs, B. and Petit, Y. (2017). "Agile Methods on Large Projects in Large Organizations." *Project Management Journal* 48 (3), 3-19.
- Jonnalagadda, G., Shafey, S., Lynah, W., Massetti, M., Maerten, P., Wiczorek, S., Landzaat, S., Probst, M., Schuster, A., Oxborough, C., Bains, R. and Bonser, M. (2017). Agile Project Delivery Confidence. PricewaterhouseCoopers International Limited.
- Kasauli, R., Knauss, E., Nilsson, A. and Klug, S. (2017). "Adding Value Every Sprint: A Case Study on Large-Scale Continuous Requirements Engineering". REFSQ Workshops, 2017.
- Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews*.
- Laanti, M. (2014). "Characteristics and Principles of Scaled Agile." 9-20.
- Larman, C. and Vodde, B. (2013). "Scaling Agile Development." *CrossTalk: The Journal of Defense Software Engineering* 26 (3), 8-12.
- Martini, A., Pareto, L. and Bosch, J. (2013). "Communication factors for speed and reuse in large-scale agile software development." In: *Proceedings of the 17th International Software Product Line Conference*. Tokyo, Japan, pp. 42-51.
- Martini, A., Pareto, L. and Bosch, J. (2016). "A multiple case study on the inter-group interaction speed in large, embedded software companies employing agile." *Journal of Software: Evolution & Process* 28 (1), 4-26.
- Miller, G. G. (2001). "The characteristics of agile software processes". 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems, 2001. IEEE.
- Moe, N. B. and Dingsøyr, T. (2017). "Emerging research themes and updated research agenda for large-scale agile development: a summary of the 5th international workshop at XP2017." In: *Proceedings of the XP2017 Scientific Workshops*. Cologne, Germany, pp. 1-4.
- Moe, N. B., Smite, D., Sablis, A., Börjesson, A.-L. and Andreasson, P. (2014). "Networking in a large-scale distributed agile project." In: *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. Torino, Italy, pp. 1-8.
- Mundra, A., Misra, S. and Dhawale, C. A. (2013). "Practical scrum-scrum team: Way to produce successful and quality software". 13th International Conference on Computational Science and Its Applications, 2013. IEEE, 119-123.
- Nyrud, H. and Stray, V. (2017). "Inter-team coordination mechanisms in large-scale agile." In: *Proceedings of the XP2017 Scientific Workshops*. Cologne, Germany, pp. 1-6.
- Paasivaara, M. (2017). "Adopting SAFe to scale agile in a globally distributed organization." In: *Proceedings of the 12th International Conference on Global Software Engineering*. Buenos Aires, Argentina, pp. 36-40.

- Paasivaara, M. and Lassenius, C. (2014). "Communities of practice in a large distributed agile software development organization - Case Ericsson." *Information & Software Technology* 56 (12), 1556-1577.
- Paasivaara, M. and Lassenius, C. (2016). "Scaling scrum in a large globally distributed organization: a case study". 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE), 2016. IEEE, 74-83.
- Paasivaara, M. and Lassenius, C. (2019). "Empower Your Agile Organization Community-Based Decision Making in Large-Scale Agile Development at Ericsson." *IEEE Software* 36 (2), 64-69.
- Paasivaara, M., Lassenius, C. and Heikkilä, V. T. (2012). "Inter-team coordination in large-scale globally distributed scrum: do scrum-of-scrums really work?" In: *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*. Lund, Sweden, pp. 235-238.
- Pries-Heje, J. and Krohn, M. M. (2017). "The SAFe way to the agile organization." In: *Proceedings of the XP2017 Scientific Workshops*. Cologne, Germany, pp. 1-3.
- Project Management Institute (2018). Pulse of the Profession 2018. Project Management Institute.
- Razzak, M. A., Richardson, I., Noll, J., Canna, C. N. and Beecham, S. (2018). "Scaling agile across the global organization: an early stage industrial SAFe self-assessment." In: *Proceedings of the 13th International Conference on Global Software Engineering*. Gothenburg, Sweden, pp. 121-130.
- Rodríguez, P., Mikkonen, K., Kuvaja, P., Oivo, M. and Garbajosa, J. (2013). "Building lean thinking in a telecom software development organization: strengths and challenges". *Proceedings of the 2013 international conference on software and system process*, 2013. ACM, 98-107.
- Rolland, K., Dingsøyr, T., Fitzgerald, B. and Stol, K.-J. (2016). "Problematizing agile in the large: alternative assumptions for large-scale agile development". 39th International Conference on Information Systems, 2016. Association for Information Systems (AIS), 1-21.
- Rolland, K. H. (2016). "Scaling Across Knowledge Boundaries: A Case Study Of A Large-Scale Agile Software Development Project." In: *Proceedings of the Scientific Workshop Proceedings of XP2016*. Edinburgh, Scotland, UK, pp. 1-5.
- Rowley, J. and Slack, F. (2004). "Conducting a literature review." *Management Research News* 27 (6), 31-39.
- Sablis, A. and Smite, D. (2016). "Agile Teams in Large-Scale Distributed Context: Isolated or Connected?" In: *Proceedings of the Scientific Workshop Proceedings of XP2016*. Edinburgh, Scotland, UK, pp. 1-5.
- Scaled Agile Inc. (2018). SAFe® 4.6 Introduction - Overview of the Scaled Agile Framework® for Lean Enterprises.
- Sundararajan, S., Bhasi, M. and Vijayaraghavan, P. K. (2014). "Case study on risk management practice in large offshore-outsourced Agile software projects." *IET Software* 8 (6), 245-257.
- Sutherland, J., Viktorov, A., Blount, J. and Puntikov, N. (2007). "Distributed scrum: Agile project management with outsourced development teams". 40th Annual Hawaii International Conference on System Sciences (HICSS'07), 2007. IEEE, 274-274.
- Tendedez, H., Ferrario, M. a. M. a. F. and Whittle, J. (2018). "Software Development and CSCW: Standardization and Flexibility in Large-Scale Agile Development." *Proceedings of the ACM on Human-Computer Interaction* 2 (CSCW), 1-23.
- Turetken, O., Stojanov, I. and Trienekens, J. J. M. (2017). "Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework." *Journal of Software: Evolution & Process* 29 (6), 1-18.
- Uludağ, Ö., Kleehaus, M., Dreyman, N., Kabelin, C. and Matthes, F. (2019). "Investigating the adoption and application of large-scale scrum at a German automobile manufacturer". *Proceedings of the 14th International Conference on Global Software Engineering*, 2019. IEEE Press, 22-29.
- Usman, M., Britto, R., Damm, L.-O. and Börstler, J. (2018). "Effort estimation in large-scale software development: An industrial case study." *Information & Software Technology* 99, 21-40.

- Van Waardenburg, G. and Van Vliet, H. (2013). "When agile meets the enterprise." *Information and Software Technology* 55 (12), 2154-2171.
- Versionone and Collabnet (2019). 13th Annual State of Agile Report. VersionOne.
- Vlaanderen, K., Van Stijn, P., Brinkkemper, S. and Van De Weerd, I. (2012). "Growing into agility: process implementation paths for scrum". International Conference on Product Focused Software Process Improvement, 2012. Springer, 116-130.
- Wale-Kolade, A. Y. (2015). "Integrating usability work into a large inter-organisational agile development project: Tactics developed by usability designers." *Journal of Systems & Software*, 54-66.
- Webster, J. and Watson, R. T. (2002). "Analyzing the past to prepare for the future: Writing a literature review." *MIS Quarterly* 26 (2), 13-23.
- Williams, L. and Cockburn, A. (2003). "Agile Software Development: It's about Feedback and Change." *Computer* 36 (6), 39.