

Identifying and Structuring Challenges in Adopting Agile and Lean Practices in Large Organizations based on a Literature Analysis

Christoph Caprano
Chair for Informatics 19
Technische Universität München (TUM)
D-85748, Garching
{caprano}@in.tum.de

Abstract—Over the last two decades, agile methods have transformed and brought unique changes to software development practice by strongly emphasizing team collaboration, customer involvement, and change tolerance. The success of agile methods for small, co-located teams has inspired organizations to increasingly apply agile practices to large-scale efforts. Since these methods are originally designed for small teams, unprecedented challenges occur when introducing them at larger scale, such as inter-team coordination and communication, dependencies with other organizational units or general resistances to changes. Compared to the rich body of agile software development literature describing typical challenges, recurring challenges of stakeholders and programs in large-scale agile development has not yet been studied through secondary studies sufficiently. With this paper, we aim to fill this gap by presenting a structured literature review on challenges in large-scale agile development. We identified 79 challenges grouped into eleven categories. The most conspicuous challenge categories were culture and mindset, software architecture, and project management.

I. INTRODUCTION

Emerging in the 1990s, agile software development methods, such as Extreme Programming (XP), Feature-Driven Development, and Scrum, have transformed and brought unprecedented changes to software development practice by strongly emphasizing change tolerance, continuous delivery, and customer involvement [1], [2]. Many enterprises are already using agile methods to maximize customer value and quality of delivered software products, but are uncertain how to introduce them at scale, since they are originally designed for small, co-located teams [1], [3]. This problem is exacerbated by the fact that the adoption of agile methods at larger scale brings new challenges with it, such as inter-team coordination and communication, dependencies with other organizational units or general resistances to changes [3], [4]. Despite these known challenges, there is an industry trend towards adopting agile methods in-the-large [3], [5].

Compared to the rich body of agile software development literature describing typical challenges (cf. [6], [7] or [8]), challenges in large-scale agile development has not yet been studied through secondary studies sufficiently [3]. Dikert et al. [3] made a first attempt to solve this problem by presenting a systematic literature review of large-scale agile transforma-

tions. They identified 35 reported challenges and 29 success factors for large-scale agile transformations. However, the presented challenges are not directly related stakeholders in order to provide appropriate proven solutions for addressing them. In our larger study, we aim to fill this gap by introducing the concept of large-scale agile development patterns and to provide best practices for recurring challenges of stakeholders and programs in large-scale agile development. Our study is inspired by the pattern-based approach to Enterprise Architecture Management (EAM) [9]. As a starting point of our study, we present our qualitative findings on stakeholder- and program-related challenges in large-scale agile development endeavors based on a structured literature review. Based on this objective, four research questions (RQ) were formulated

- *RQ1: Which stakeholders exist in large-scale agile development endeavors?*
- *RQ2: What are challenges of stakeholders and programs in large-scale agile development efforts?*
- *RQ3: Which challenge categories are the most salient in large-scale agile development?*
- *RQ4: What are generalizable findings on stakeholder- and program-related challenges in large-scale agile development endeavors?*

A. Research methodology

The goal of the literature review is to identify challenges of stakeholders and programs in large-scale agile development. To identify relevant material in order to achieve this goal and to ensure the rigor and relevance of the research, we applied a structured literature review approach as recommended by Brocke et al. [10] that consists of four phases (see Fig. 1). In the first phase, we defined the review scope and formulated adequate research questions about challenges in large-scale agile development. In the second phase, we identified key concepts by concept mapping, which also provided us the opportunity to obtain relevant search terms: *Agile and Lean Software Engineering*, *Large-Scale Agile Development*, *Agile Transformation*, and *Challenges, Concerns and Problems*. These search terms were used in the subsequent literature search in the third phase. We examined a range of different Information

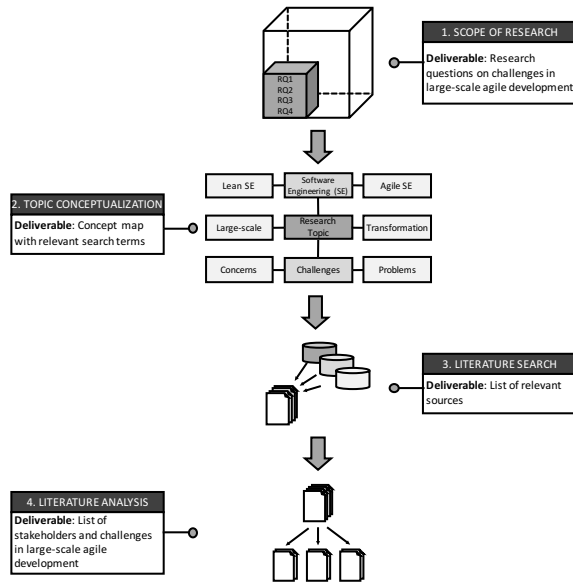


Fig. 1. Research approach and deliverables.

Systems journals, conference proceedings using ACM Digital Library, IEEEExplore, Scopus, and Web of Science. Having compiled the aforementioned list of search terms, we then used them in electronic full-text search queries. Initially, 67 sources were identified as relevant, given their focus on the topic, after analyzing a total of 560 sources (title, abstract, and outline). Additionally, we conducted a backward search, resulting in additional 6 sources. In total, we obtained 73 relevant sources. In the fourth phase, we coded the primary studies using a deductive approach as proposed by Cruzes and Dybå [11]. We established an a priori list of codes inspired by the EAM pattern language elements [9], which includes stakeholders, challenges, methodology patterns, architecture principles, viewpoint patterns, and anti-patterns. In the initial step, we started with the actual coding of the data structured by the aforementioned code categories. During the coding, we also identified the relationships between the codes of the different code families. Particularly, we related challenges to respective stakeholders and solutions, such as methodology patterns or architecture principles. Based on this structure, we can for instance determine typical concerns of solution architects and how they are trying to address them. After creating preliminary codes, we refined and consolidated our codes by merging related ones and removing duplicates. In the final step, we grouped related challenges into eleven categories. Table I presents a description of the codes families and the final state of the coding. Note that in this paper, we only discuss the results related to challenges and stakeholders, and that it as such forms a part of a larger study. In our larger study, we aim to introduce the concept of large-scale agile development patterns, which builds on and extends the idea of the proven pattern-based approach to EAM [9]. The aim of this new pattern language is to address typical problems of stakeholders and programs in large-scale agile development

endeavors.

The remainder of this paper is structured as follows. In Section II, we provide an overview of related works describing challenges and success stories of large-scale agile development endeavors. In Section III, we present our findings on large-scale agile development challenges identified in the literature. We discuss the main findings in Section IV before concluding the paper with a summary of our results and remarks on future research in Section V.

II. RELATED WORK

Dikert et al. [3] conducted a systematic literature review of industrial large-scale agile transformations focusing on reported challenges and success factors in the transformation. 47 out of 117 relevant papers were selected to obtain 35 challenges and 29 success factors for agile adoption. They grouped the challenges in nine categories and the success factors in eleven categories. Paasivaara and Lassenius [12] validated and deepened these findings with a pilot study. The result is an improved and weighted version of the success factors and challenges. However, there is no relationship to stakeholders that are affected by these concerns. Viswanath [13] observed more than 400 employees in a company during their five years long lean transformation. The employees were analyzed while facing challenges, pitfalls and success factors according to three dimensions: Process, Product and People. Thereby, Viswanath showed that every stakeholder in a company is involved in the lean transformation. Nevertheless, detailed relations of challenges to stakeholders is also missing in this paper.

Bjarnason et al. [14] reported that agile transformations does not affect only software developers but also other disciplines like requirements engineering. The challenges of over-scoping and communication gaps can be addressed with agile approaches. Some of the to be faced challenges are similar to the traditional way but the transformation cause also new challenges like the assurance of the right balance of agility and stability. Bjarnason et al. also presented that requirements engineering is affected by the agile transformation and have to face new challenges. Unfortunately, there is right now no collection of stakeholder specific challenges that directly mention stakeholders like requirement engineers.

Angelov et al. [15] presented in their study that also architects face challenges in large-scale development. These stakeholders have mostly concerns with the autonomous teams and the product owner. In their contribution, they conducted a case-study of six Dutch companies to show challenges, pitfalls and success factors of applying architecture in large-scale development. Thereby, they differentiated between different architecture roles and their different challenges during the case-study. Similar to Bjarnason et al. [14], the shown challenges are not included in a collection of stakeholder specific challenges. All in all, stakeholder-specific challenges are distributed over multiple papers and there is no paper that gives an overview about the stakeholder-specific challenges.

TABLE I
OVERVIEW OF CODE FAMILIES AND CODES

Code family	Description	Examples	# Identified elements	Codes
Stakeholders	A person with an interest or concern in a large-scale agile development endeavor	Product owner, scrum master, software architect	14	770
Challenges	Challenges describe interests of programs or stakeholders that have certain goals in large-scale agile development	Ensuring that non-functional requirements are considered by the development team	79	286
<i>M-Patterns</i>	Methodology patterns (M-Patterns) are defined as concrete steps that are performed to address recurring concerns of large-scale agile development programs and stakeholder	Scrum of scrums, community of practices, creating an architectural runway	122	237
<i>Architecture Principles</i>	Architecture principles define the underlying general rules and guidelines for the use and deployment of all IT resources and assets across the enterprise	Loose coupling of systems or services, reuse of functionalities, buy before make	4	5
<i>V-Patterns</i>	Viewpoint patterns (V-Patterns) are defined as documentations of proven practices to recurring problems for specific contexts in form of viewpoints for the creation of views	Burndown chart, context map, pulse chart	9	12
<i>Anti-Patterns</i>	Anti-patterns detail on typical mistakes in large-scale agile development, and present revised solutions, which help pattern users to prevent these pitfalls	Don't put individual goals over team goals, don't adopt all agile practices in one go, don't overshoot coordination meetings	17	68
Total				1378

TABLE II
IDENTIFIED STAKEHOLDERS

ID	Name	# Documents
S-2	Enterprise Architect	3
S-3	Program Manager	4
S-4	Business Analyst	3
S-5	Support Engineer	2
S-7	Development Team	50
S-8	Product Owner	33
S-11	UX Expert	1
S-13	Agile Coach	4
S-14	Solution Architect	2
S-16	Test Team	18
S-17	Software Architect	21
S-19	Scrum Master	30
S-20	Portfolio Manager	2
S-42	Product Manager	13

III. CHALLENGES IN LARGE-SCALE AGILE DEVELOPMENT PROGRAMS

A. Stakeholders in large-scale agile development

In our structured literature review, we identified 40 stakeholder roles that are involved in large-scale agile development. Many of them are either already present in traditional software development or a synonym for another role. Therefore, we consolidated the 40 roles to 14 stakeholder roles (see Table II). One example of this consolidation is the role of the Chief Architect [16] which has been merged with the Enterprise Architect because of their similar areas of responsibilities. The five most important stakeholder roles found in literature were thereby the following

- Development team,

- product owner,
- test team,
- scrum master and
- software architect.

It is remarkable that the role of the architect is mentioned in many papers even if the role of the architect is not included in many agile methods, such as XP or Scrum [15]. Other roles, like the product manager or portfolio manager are added in large-scale agile development to support the management of large programs.

B. Challenges in large-scale agile development

We also observed in the literature review that stakeholder roles struggle with challenges that either newly arose or at least are strengthened by large-scale agile development [17] [15]. Altogether, we identified 79 challenges of which 41 newly arose by large-scale agile development and 38 are strengthened by large-scale agile development (see Table III and Table IV). In the following, we will introduce the five most frequent identified challenges.

Coordinating multiple agile teams that work on the same product. We observed in 15 papers that the most frequent challenge is the coordination of multiple agile teams. If the teams also work on the same product, the coordination and communication between the teams seems to be challenging in large-scale agile development.

Considering integration issues and dependencies with other subsystems and teams. The second most frequent concern shows that teams struggle with the integration of their product increment with other subsystems. One issue is the dependency management to other subsystems or teams that

seems to be challenging. This concern was observed in 14 papers.

Coordinating geographically distributed agile teams. Another problem, which was observed in eight papers, is the distribution of teams or team members. The coordination of them is perceived as very difficult in a distributed agile software development setting. Especially the distance between people or teams as well as the number of time zones makes it harder to coordinate agile teams.

Dealing with doubts in people about changes. The change to agile software development is not supported by everyone in a company. Many people don't want to change their way of working and don't trust the agile way. They have to be integrated and convinced but we observed in seven papers that this is perceived as challenging.

Facilitating shared context and knowledge. In agile software development, many teams are cross-functional because it supports their independence way of working. Nevertheless, this hinders the communication between people and their exchange of knowledge because in contrast to traditional functional departments, the people with the same interests do not work together. This concern is mentioned in seven papers.

C. Challenges categorized by stakeholders

The identified challenges are either program-specific or are faced by specific stakeholders. Therefore, we analyzed which challenge concerns which stakeholder. Table II shows the identified stakeholder groups and the relationship to their concerns is illustrated in Table III and IV. In the following, we will introduce the concerns of the five most frequent observed stakeholders.

Development team: The most frequent mentioned stakeholder is the development team. Nevertheless, we identified only three challenges for them. Compared to other stakeholders, the development team has to face less challenges in large-scale agile development. The main concerns of this stakeholder group is mostly related to self-organization, documentation of their work in a lightweight, and the estimation of user stories. All of their concerns still exist in agile development but they are intensified in large-scale agile development.

Product owner: Another established agile role is the role of the product owner. The product owner is mentioned in 33 papers. We observed for this stakeholder 14 challenges. The product owner has to provide precise requirement specifications to the development team as well as has to share the common vision of the to be developed product with other stakeholders. Thereby, one responsibility is to split complex requirements into smaller requirements. This work is perceived as challenging because in the past, the requirements were created in a very detailed manner. Another concern of the product owner is to facilitate communication between agile teams and teams using traditional practises. This includes for instance the communication for supporting functions like human resources or sales [26].

Scrum master: One typical concern of the scrum master is to remove impediments of the development team. If the number of teams and team distribution increases, the scrum master has the responsibility to coordinate the distributed teams, synchronize their working hours, ensure the team cohesion at different locations, and to facilitate the participation of agile teams at cross-shore meetings. In addition, the scrum master has to ensure that stakeholders trust in practicing agile values and principles. Furthermore, the scrum master prevents potential threats for the development team.

Software architect: The software architect is the fourth most mentioned stakeholder in our structured literature review. For the software architect, we identified seven challenges that all newly arose from large-scale agile development. The main concern the software architect is confronted with is the management of technical debts. The software architect also has to ensure that architectural decisions are included in the development process. Thereby, the software architect has to create a proper upfront architecture design of the system. In addition, the software architect has to ensure that the non-functional requirements are implemented. Last but not least, the software architect supports agile team by presenting them possible solution on how to develop and maintain legacy systems.

Test team: The test team was observed in 18 documents. We identified for the test team only three challenges, which is similar to the development team. The main concerns of the test team is to establish and create understandable automated tests that can be referenced to requirements. This helps to show and maintain the correct implementation of the requirements.

D. Challenges categorized in topics

We grouped the 79 identified challenges into eleven challenge categories. These categories are illustrated in Table 2. The five topics with the most concerns are the following:

Culture & mindset: The topic with the most challenges is culture & mindset. Thereby, this topic illustrates that it is challenging to convince all of the involved stakeholders for the agile practices. This includes external as well as internal stakeholders, like the team itself. The team also has concerns according to the establishment of a culture of continuous improvement and the creation of a team spirit and trust among team members. In total, this topic includes 14 challenges.

Software architecture: Software architecture includes ten large-scale agile development challenges. Thereby, it is the topic with the second most challenges. This topic contains similar challenges as the challenges of the previously mentioned software architect. The main concern is to ensure that non-functional requirements and architectural decisions are taken into account by the development team.

Requirements engineering: Nine challenges can be assigned to the topic of requirements engineering. The process of the requirements engineering is mostly done by the product owner and therefore, the concerns are similar to the challenges of this stakeholder role. The requirements topic contains

TABLE III
LARGE-SCALE AGILE DEVELOPMENT CHALLENGES

ID	Name	Category	Novelty	Affected stakeholders or program	Origin
C-1	Ensuring that non-functional requirements are considered by the development team	Software-Architecture	yes	Software Architect, Solution Architect	[18], [12], [19], [20], [21], [3]
C-2	Creating precise requirement specifications for the development team	Requirements Engineering	no	Product Owner	[18], [12], [22], [23], [3]
C-3	Managing and integrating heterogenous subsystems of different development teams	Software-Architecture	yes	Solution Architect	[18], [24], [3]
C-4	Defining a lightweight formal review process for new technologies	Enterprise Architecture	yes	Enterprise Architect	[25]
C-6	Facilitating communication between agile teams and other teams using traditional practices	Communication & Coordination	yes	Epic Owner, Product Owner	[25], [26], [23], [3]
C-7	Managing dependencies to other existing environments	Enterprise Architecture	yes	Enterprise Architect	[25], [19], [23], [3]
C-8	Obtaining management buy-in	Culture & Mindset	no	Program specific	[25], [27], [12], [26], [3]
C-10	Dealing with black and white mindsets	Culture & Mindset	no	Agile Coach	[25], [3]
C-11	Dealing with office politics	Culture & Mindset	no	Program specific	[25]
C-12	Dealing with closed mindedness	Culture & Mindset	no	Agile Coach	[25], [3]
C-13	Coordinating multiple agile teams that work on the same product	Communication & Coordination	yes	Program Manager	[28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [19], [39], [24], [3]
C-14	Aligning and communicating architectural decisions	Software-Architecture	yes	System Architect, Solution Architect	[15], [24], [3]
C-15	Dealing with higher-level management interferences	Culture & Mindset	no	Scrum Master	[15], [24]
C-18	Demonstrating the value of architecting	Software-Architecture	yes	Software Architect	[15], [40]
C-21	Fostering technical excellence	Software-Architecture	yes	Software Architect	[15]
C-24	Managing and sharing knowledge about system components and their dependencies with stakeholders	Enterprise Architecture	yes	Enterprise Architect	[24], [34], [3]
C-25	Finding the right balance between architectural improvements and business value	Software-Architecture	yes	Software Architect	[24], [21], [20], [19], [40], [41]
C-27	Balancing short-term and long-term goals	Requirements Engineering	no	Product Manager	[42], [12], [38], [3]
C-28	Considering integration issues and dependencies with other subsystems and teams	Software-Architecture	yes	Solution Architect	[32], [20], [24], [43], [41], [19], [21], [44], [45], [36], [37], [30], [12], [3]
C-29	Dealing with increased efforts by establishing inter-team communication	Communication & Coordination	yes	Program specific	[32], [41]
C-31	Dealing with lacking sense of ownership responsibilities for developed services	Culture & Mindset	yes	Program specific	[32], [21]
C-32	Communicating business requirements to development teams	Requirements Engineering	no	Product Owner	[46], [38], [19]
C-33	Providing sufficient tools and infrastructure for remote communications	Tooling	no	Program specific	[47], [48], [49], [50], [39]
C-34	Encouraging development teams to talk about tasks and impediments	Culture & Mindset	no	Agile Coach, Scrum Master	[47]
C-35	Facilitating agile teams to participate at cross-shore meetings	Geographical Distribution	yes	Scrum Master	[47], [49], [3]
C-36	Synchronizing working hours of cross-shore agile teams	Geographical Distribution	yes	Scrum Master	[47], [48], [3]
C-38	Dealing with geographical distance between agile teams	Geographical Distribution	yes	Program specific	[47], [33], [48]
C-39	Dealing with lacking team cohesion at different locations	Geographical Distribution	yes	Scrum Master	[47], [48], [21]
C-40	Facilitating shared context and knowledge	Knowledge Management	no	Program specific	[47][47], [51], [13], [41], [52], [34], [21]
C-41	Managing technical debts	Software-Architecture	no	Software Architect	[53], [24], [13], [31], [21], [25], [32]
C-42	Sharing common vision	Knowledge Management	yes	Program Manager, Product Owner	[54], [41], [13], [55], [3]
C-43	Building trust of stakeholders in agile practices	Culture & Mindset	no	Chief Scrum Master	[54], [53], [3]
C-44	Creating a proper upfront architecture design of the system	Software-Architecture	yes	Software Architect	[20], [43], [56], [27], [24]
C-45	Ensuring that agile teams adhere to architecture-related activities	Enterprise Architecture	yes	Enterprise Architect	[56], [24]
C-46	Ensuring the reuse of enterprise assets	Enterprise Architecture	yes	Enterprise Architect	[56], [27], [24]
C-47	Defining clear and visible priorities	Requirements Engineering	no	Product Owner	[31], [57], [22]
C-48	Providing agile teams appropriate automation and scalable infrastructure	Tooling	no	Enterprise Architect	[21], [29]
C-49	Establishing automated testing	Quality Assurance	no	Test Team	[12], [13], [3]
C-50	Writing understandable automated tests	Quality Assurance	no	Test team	[58]
C-51	Ensuring traceability of tests and requirements	Quality Assurance	no	Test team	[58], [3]

TABLE IV
LARGE-SCALE AGILE DEVELOPMENT CHALLENGES (CONTINUED)

ID	Name	Category	Novelty	Affected stakeholders or program	Origin
C-52	Establishing a common scope for different stakeholder groups	Knowledge Management	yes	Program specific	[44], [50], [31], [47]
C-53	Making a cost and schedule estimation	Project Management	no	Product Owner, Product Manager, Program Manager	[19], [44]
C-54	Creating lightweight documentation	Knowledge Management	no	Development team	[19], [23], [27]
C-55	Establishing requirements verification	Requirements Engineering	no	Product Owner	[19]
C-56	Eliciting and refining requirements of end users	Requirements Engineering	no	Product Owner	[19], [38], [23], [12], [3]
C-58	Defining high-level requirements a.k.a. epics	Requirements Engineering	yes	Portfolio Manager, Product Owner	[12]
C-59	Measuring the success of the large-scale agile development program	Project Management	yes	Product Owner	[27]
C-60	Creating a teamwork centric rewarding model	Project Management	no	Program specific	[12], [3]
C-61	Dealing with increasing workload of key stakeholders	Project Management	yes	Program specific	[41], [44], [19], [21], [3]
C-62	Considering required competencies when assigning teams to tasks	Project Management	yes	Program specific	[41]
C-64	Defining clear roles and responsibilities	Project Management	no	Program specific	[22], [12]
C-65	Decomposing agile teams in smaller independent teams	Enterprise Architecture	yes	Program Manager, Enterprise Architect	[44], [59]
C-66	Dealing with decreased predictability	Project Management	no	Program specific	[27]
C-68	Dealing with loss of management control	Culture & Mindset	no	Program specific	[27], [41]
C-69	Establishing self-organization	Communication & Coordination	no	Development team	[57], [34], [31], [21], [3]
C-70	Facilitating standardization across agile teams	Enterprise Architecture	yes	Enterprise Architect	[60], [12], [3]
C-71	Dealing with incorrect practices of agile development	Methodology	no	Agile Coach	[43], [38], [12], [53], [61], [34], [21]
C-72	Creating team spirit and trust among agile teams	Culture & Mindset	yes	Program specific	[33], [53], [31], [21]
C-73	Establishing a culture of continuous improvement	Culture & Mindset	no	Scrum Master, Agile Coach	[38], [62], [59]
C-74	Establishing a common understanding of agile thinking and practices	Methodology	yes	Agile Coach	[12], [3]
C-75	Empowering agile teams to make decisions	Culture & Mindset	no	Program specific	[31]
C-76	Forming and managing autonomous teams	Communication & Coordination	yes	Program specific	[12]
C-77	Applying agile practices for developing or maintaining legacy systems	Software-Architecture	yes	Software Architect	[18], [13], [27]
C-78	Creating and estimating user stories	Requirements Engineering	no	Product Owner, Development Team	[12], [3]
C-79	Splitting large and complex requirements into smaller requirements	Requirements Engineering	yes	Product Owner, Program Manager	[63], [21], [12], [13], [3]
C-80	Dealing with unplanned requirements and risks	Project Management	no	Program Manager, Product Owner, Product Manager	[23], [41], [21]
C-81	Coordinating tests and deployment with external parties	Quality Assurance	no	Test team, Development Team	[41]
C-84	Coordinating geographically distributed agile teams	Geographical Distribution	yes	Scrum Master	[64], [21], [48], [12], [23], [38], [31], [3]
C-85	Dealing with cultural differences between cross-shore agile teams	Geographical Distribution	yes	Scrum Master	[33], [50]
C-87	Establishing a lightweight review process for adopting new technologies	Enterprise Architecture	yes	Enterprise Architect	[29]
C-88	Rearranging physical spaces	Tooling	no	Scrum Master	[12], [49], [3]
C-89	Building an effective coaching model	Methodology	no	Agile Coach	[65]
C-90	Enforcing customer involvement	Culture & Mindset	no	Product Owner	[19], [22], [27]
C-91	Dealing with internal silos	Knowledge Management	yes	Program specific	[12], [38], [59], [24], [3]
C-92	Dealing with fixed price contracts in agile software development	Project Management	no	Product Manager, Program Manager	[66], [27]
C-93	Synchronizing sprints in the large-scale agile development program	Communication & Coordination	yes	Scrum Master	[60]
C-94	Explaining requirements to stakeholders	Communication & Coordination	no	Development Team	[55], [38]
C-95	Dealing with communication gaps with stakeholders	Communication & Coordination	yes	Program specific	[19], [50], [59]

thereby mainly challenges that deals with the requirements specification and requirements communication.

Project management: Project management (PM) contains eight challenges. They mostly represent the problem of an efficient management of the available human resources. This includes the workload management of key stakeholders. These stakeholders are often experts in a special area and are assigned to multiple agile teams. Thereby, it is likely that they are overloaded and it is thereby required to prevent work overload for them. Another PM concern is the management of unplanned requirements and risks. Even though agile software development supports scenarios with high probability of changing requirements, this concern still seems to be challenging. The increasing number of stakeholders strengthens this effect. Other PM challenges for large-scale agile development are the creation of a team-centric rewarding model and the creation of clear roles and responsibilities. In agile software development, it is required to get rid of the individual goal rewarding models. In addition, the creation of clear roles and responsibilities supports the agile success because stakeholders perform better when they know their role and responsibilities in a program or team setting.

Communication & coordination: The topic with the fifth most challenges is communication & coordination with eight challenges: it turned out that when the number of teams, their distribution and the number of external stakeholder increases, the communication and coordination effort also increases. These challenges include the problem of coordinating multiple teams that work on the same product. Another challenge is the establishment of self-organization in agile teams. Further concerns deal with communication gaps between the stakeholders as well as managing the increased effort for inter-team communication.

IV. DISCUSSION

A. Key findings

Let us now reflect on the four research questions described in Section I.

RQ1: Which stakeholders exist in large-scale agile development endeavors? In the literature review, we observed 40 different stakeholder roles in large-scale agile development endeavors. We consolidated them to 14 final stakeholder roles, which are listed Table II). The stakeholder roles include roles from agile software development as well as new roles, like software architects or portfolio managers. It is remarkable that every traditional software development stakeholder role can be mapped to an agile development role. The stakeholders are thereby mainly responsible for similar tasks as in traditional software development. The only difference is mostly a change in the way of working.

RQ2: What are challenges of stakeholders and programs in large-scale agile development efforts? We identified 79 challenges for large-scale agile development (see Table III, IV). These challenges can be either program-specific or are to be faced by specific stakeholders. Thereby, we assigned the

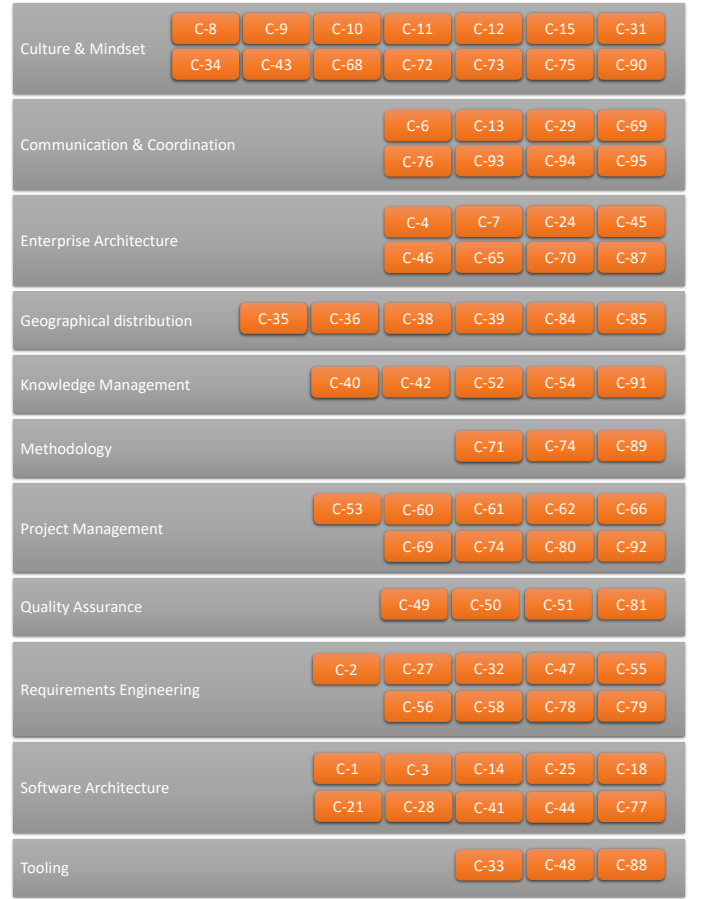


Fig. 2. 79 challenges grouped in eleven topics

challenges either to the in RQ1 observed stakeholder roles or marked them as program-specific.

RQ3: Which challenge categories are the most salient in large-scale agile development? The identified challenges can be categorized in eleven topics. We assigned one topic per challenge and visualized the results in Figure 2.

RQ4: What are generalizable findings on stakeholder- and program-related challenges in large-scale agile development endeavors? Architecture becomes more important the more complex the task or system is. It is remarkable that the software architect is at the fifth position after the traditional roles even if the role of the architect is not included in many agile methods, such as XP or Scrum [15]. Furthermore, more than 20% of the challenges are related to architecture-related topics. All of these challenges newly arose with large-scale agile development. New stakeholder roles are involved when scaling agile development. Although, the role of the architect was not intended in agile software development, because it only contains the role of a product owner, scrum master and the development team and additional ones are not mentioned [67]. Nevertheless, we observed in the literature analysis further roles like software and enterprise architects or product managers. Scaling agile development entails new communication and coordination challenges. The additional stakeholder roles

help to manage big software programs. This includes also the management of multiple agile teams. In the literature review, we identified eight communication and coordination challenges and 75% of them newly arose from large-scale agile development. *Challenges in agile development may still exist in large-scale agile development.* We identified 79 challenges in large-scale agile development. 38 of them still exist in large-scale agile development. These challenges are typical for agile development and are reinforced by large-scale agile development. *Stakeholders that are successfully isolated by the scrum master from external influences have less concerns in large-scaled agile development.* Only 7% of the observed challenges are either challenges for the development or test team. This is quite low compared to the number of challenges of the other stakeholder roles. For instance, the product owner has to face 20% of the observed challenges. Furthermore, the top challenge topics are inter-team coordination and communication problems as well as architecture related issues. Stakeholders that are isolated by the scrum master from external influences are not affected by these challenges and face thereby less challenges in large-scale agile development.

B. Limitations

This paper has a few limitations, which should be mentioned at this point: First, although, we spent much time and effort into developing a suitable search string and conducted a structured database search, there is still a certain chance that not all important contributions have been identified. We found additional literature through a backward search of the analyzed papers in the literature search process. Some relevant studies might have evaded our attention in spite of our best efforts. Second, the initial coding procedure was conducted by only one researcher, which might have led to biased classifications. It might have been better if two researchers had been involved working on a pair coding mode from the beginning.

V. CONCLUSION AND FUTURE WORK

In this study, we presented a structured literature review on recurring challenges of stakeholders and programs in large-scale agile development. We analyzed 73 papers, in order to describe reported challenges for large-scale agile development endeavors. In total, 79 challenges were identified and grouped into eleven challenge categories, which are *culture & mindset, communication & coordination, enterprise architecture, geographical distribution, knowledge management, methodology, project management, quality assurance, requirements engineering, software architecture*, and *tooling*. We will extend our preliminary study by collecting data from our large-scale agile development workshops and case studies with industry partners. In parallel, we will perform a structured survey among companies in Germany to demonstrate the applicability of our large-scale agile pattern language, which provides the structure for documenting practice-proven solutions to recurring large-scale agile development problems. After a huge data collection and evaluating the new pattern language, we

will publish the Large-Scale Agile Pattern Catalog containing patterns and concerns.

REFERENCES

- [1] P. Kettunen, "Extending software project agility with new product development enterprise agility," *Software Process: Improvement and Practice*, vol. 12, no. 6, pp. 541–548, 2007.
- [2] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," 2012.
- [3] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87 – 108, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121216300826>
- [4] M. Alqudah and R. Razali, "A review of scaling agile methods in large software development," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, no. 6, pp. 28–35, 2016.
- [5] VersionOne, "12th annual state of agile report," VersionOne, Tech. Rep., 2018.
- [6] E. Hossain, M. A. Babar, and H.-y. Paik, "Using scrum in global software development: a systematic literature review," in *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*. Ieee, 2009, pp. 175–184.
- [7] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in human behavior*, vol. 51, pp. 915–929, 2015.
- [8] W. R. Fitriani, P. Rahayu, and D. I. Sensuse, "Challenges in agile software development: A systematic literature review," in *Advanced Computer Science and Information Systems (ICACSIS), 2016 International Conference on*. IEEE, 2016, pp. 155–164.
- [9] A. W. Schneider and F. Matthes, "Evolving the eam pattern language," in *Proceedings of the 20th European Conference on Pattern Languages of Programs*. ACM, 2015, p. 45.
- [10] J. Vom Brocke, A. Simons, B. Niehaves, K. Riemer, R. Plattfaut, and A. Cleven, "Reconstructing the giant: On the importance of rigour in documenting the literature search process," in *ECIS*, vol. 9, 2009, pp. 2206–2217.
- [11] D. S. Cruzes and T. Dyba, "Recommended steps for thematic synthesis in software engineering," in *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on*. IEEE, 2011, pp. 275–284.
- [12] M. Paasivaara and C. Lassenius, "Scaling scrum in a large globally distributed organization: A case study," in *Global Software Engineering (ICGSE), 2016 IEEE 11th International Conference on*. IEEE, 2016, pp. 74–83.
- [13] U. Viswanath, "Lean transformation: Adapting to the change, factors for success and lessons learnt during the journey: A case study in a multi location software product development team," in *Proceedings of the 9th India Software Engineering Conference*, ser. ISEC '16. New York, NY, USA: ACM, 2016, pp. 156–162. [Online]. Available: <http://doi.acm.org/10.1145/2856636.2856657>
- [14] E. Bjarnason, K. Wnuk, and B. Regnell, "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*, ser. AREW '11. New York, NY, USA: ACM, 2011, pp. 31–35.
- [15] S. Angelov, M. Meesters, and M. Galster, "Architects in scrum: What challenges do they face?" 11 2016, pp. 229–237.
- [16] A. Martini, J. Bosch, and M. Chaudron, "Architecture technical debt: Understanding causes and a qualitative model," in *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*, Aug 2014, pp. 85–92.
- [17] S. Buckl, J. Lankes, C. M. Schweda, and F. Matthes, "Eam pattern catalog v1," 2005.
- [18] B. Boehm and R. Turner, "Management challenges to implementing agile processes in traditional development organizations," *IEEE Software*, vol. 22, no. 5, pp. 30–39, Sept 2005.
- [19] K. H. Rolland, "'desperately' seeking research on agile requirements in the context of large-scale agile projects," in *Scientific Workshop Proceedings of the XP2015*, ser. XP '15 workshops. New York, NY, USA: ACM, 2015, pp. 5:1–5:6. [Online]. Available: <http://doi.acm.org/10.1145/2764979.2764984>

- [20] M. A. Babar, "An exploratory study of architectural practices and challenges in using agile software development approaches," in *2009 Joint Working IEEE/IFIP Conference on Software Architecture European Conference on Software Architecture*, Sept 2009, pp. 81–90.
- [21] M. S. Roopa, C. Sankarabubiah, and V. S. Mani, "Usable software at the end of each takt: A milestone in the lean transformation of a globally distributed software development team," in *Proceedings of the 12th International Conference on Global Software Engineering*, ser. ICGSE '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 116–120. [Online]. Available: <https://doi.org/10.1109/ICGSE.2017.9>
- [22] H. Ayed, B. Vanderose, and N. Habra, "Supported approach for agile methods adaptation: An adoption study," in *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, ser. RCoSE 2014. New York, NY, USA: ACM, 2014, pp. 36–41. [Online]. Available: <http://doi.acm.org/10.1145/2593812.2593820>
- [23] M. Budwig, S. Jeong, and K. Kelkar, "When user experience met agile: A case study," in *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '09. New York, NY, USA: ACM, 2009, pp. 3075–3084. [Online]. Available: <http://doi.acm.org/10.1145/1520340.1520434>
- [24] A. Martini and J. Bosch, "The danger of architectural technical debt: Contagious debt and vicious circles," in *2015 12th Working IEEE/IFIP Conference on Software Architecture*, May 2015, pp. 1–10.
- [25] A. Mahanti, "Challenges in enterprise adoption of agile methods," 2006.
- [26] J. Pries-Heje and M. M. Krohn, "The safe way to the agile organization," in *Proceedings of the XP2017 Scientific Workshops*, ser. XP '17. New York, NY, USA: ACM, 2017, pp. 18:1–18:3. [Online]. Available: <http://doi.acm.org/10.1145/3120459.3120478>
- [27] P. Rodríguez, J. Markkula, M. Oivo, and K. Turula, "Survey on agile and lean usage in finnish software industry," in *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '12. New York, NY, USA: ACM, 2012, pp. 139–148. [Online]. Available: <http://doi.acm.org/10.1145/2372251.2372275>
- [28] K. Rautiainen, J. von Schantz, and J. Vahaniitty, "Supporting scaling agile with portfolio management: Case paf.com," in *2011 44th Hawaii International Conference on System Sciences*, Jan 2011, pp. 1–10.
- [29] R. P. Maranzato, M. Neubert, and P. Herculano, "Moving back to scrum and scaling to scrum of scrums in less than one year," in *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*, ser. OOPSLA '11. New York, NY, USA: ACM, 2011, pp. 125–130. [Online]. Available: <http://doi.acm.org/10.1145/2048147.2048186>
- [30] T. Dybå and T. Dingsøyr, "Agile project management: From self-managing teams to large-scale development," in *Proceedings of the 37th International Conference on Software Engineering - Volume 2*, ser. ICSE '15. Piscataway, NJ, USA: IEEE Press, 2015, pp. 945–946. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2819009.2819222>
- [31] R. K. Gupta, P. Manikreddy, and K. C. Arya, "Pragmatic scrum transformation: Challenges, practices & impacts during the journey a case study in a multi-location legacy software product development team," in *Proceedings of the 10th Innovations in Software Engineering Conference*, ser. ISEC '17. New York, NY, USA: ACM, 2017, pp. 147–156. [Online]. Available: <http://doi.acm.org/10.1145/3021460.3021478>
- [32] E. Moore and J. Spens, "Scaling agile: Finding your agile tribe," in *Agile 2008 Conference*, Aug 2008, pp. 121–124.
- [33] R. Vivian, H. Tarmazdi, K. Falkner, N. Falkner, and C. Szabo, "The development of a dashboard tool for visualising online teamwork discussions," in *Proceedings of the 37th International Conference on Software Engineering - Volume 2*, ser. ICSE '15. Piscataway, NJ, USA: IEEE Press, 2015, pp. 380–388. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2819009.2819070>
- [34] N. B. Moe, H. H. Olsson, and T. Dingsøyr, "Trends in large-scale agile development: A summary of the 4th workshop at xp2016," in *Proceedings of the Scientific Workshop Proceedings of XP2016*, ser. XP '16 Workshops. New York, NY, USA: ACM, 2016, pp. 1:1–1:4. [Online]. Available: <http://doi.acm.org/10.1145/2962695.2962696>
- [35] T. Dingsøyr, K. Rolland, N. B. Moe, and E. A. Seim, "Coordination in multi-team programmes: An investigation of the group mode in large-scale agile software development," *Procedia Computer Science*, vol. 121, pp. 123 – 128, 2017, cENTERIS 2017 - International Conference on ENTERprise Information Systems / ProjMAN 2017 - International Conference on Project MANAGEMENT / HCist 2017 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877050917322081>
- [36] K. Crowston, K. Chudoba, M. B. Watson-Manheim, and P. Rahmati, "Inter-team coordination in large-scale agile development: A test of organizational discontinuity theory," in *Proceedings of the Scientific Workshop Proceedings of XP2016*, ser. XP '16 Workshops. New York, NY, USA: ACM, 2016, pp. 2:1–2:5. [Online]. Available: <http://doi.acm.org/10.1145/2962695.2962697>
- [37] S. Bick, A. Scheerer, and K. Spohrer, "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*, ser. XP '16 Workshops. New York, NY, USA: ACM, 2016, pp. 4:1–4:5. [Online]. Available: <http://doi.acm.org/10.1145/2962695.2962699>
- [38] M. Paasiavaara, "Adopting safe to scale agile in a globally distributed organization," in *Proceedings of the 12th International Conference on Global Software Engineering*, ser. ICGSE '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 36–40. [Online]. Available: <https://doi.org/10.1109/ICGSE.2017.15>
- [39] H. Nyruud and V. Stray, "Inter-team coordination mechanisms in large-scale agile," in *Proceedings of the XP2017 Scientific Workshops*, ser. XP '17. New York, NY, USA: ACM, 2017, pp. 16:1–16:6. [Online]. Available: <http://doi.acm.org/10.1145/3120459.3120476>
- [40] P. Abrahamsson, M. A. Babar, and P. Kruchten, "Agility and architecture: Can they coexist?" *IEEE Software*, vol. 27, no. 2, pp. 16–22, March 2010.
- [41] J. E. Hannay and H. C. Benestad, "Perceived productivity threats in large agile development projects," in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '10. New York, NY, USA: ACM, 2010, pp. 15:1–15:10. [Online]. Available: <http://doi.acm.org/10.1145/1852786.1852806>
- [42] M. Laanti, "Implementing program model with agile principles in a large software development organization," in *2008 32nd Annual IEEE International Computer Software and Applications Conference*, July 2008, pp. 1383–1391.
- [43] B. Murphy, C. Bird, T. Zimmermann, L. Williams, N. Nagappan, and A. Begel, "Have agile techniques been the silver bullet for software development at microsoft?" in *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, Oct 2013, pp. 75–84.
- [44] M. Kircher and P. Hofman, "Combining systematic reuse with agile development: Experience report," in *Proceedings of the 16th International Software Product Line Conference - Volume 1*, ser. SPLC '12. New York, NY, USA: ACM, 2012, pp. 215–219. [Online]. Available: <http://doi.acm.org/10.1145/2362536.2362566>
- [45] D. Rosenberg, B. Boehm, B. Wang, and K. Qi, "Rapid, evolutionary, reliable, scalable system and software development: The resilient agile process," in *Proceedings of the 2017 International Conference on Software and System Process*, ser. ICSSP 2017. New York, NY, USA: ACM, 2017, pp. 60–69. [Online]. Available: <http://doi.acm.org/10.1145/3084100.3084107>
- [46] D. Broschinsky and L. Baker, "Using persona with xp at landesk software, an avocent company," in *Agile 2008 Conference*, Aug 2008, pp. 543–548.
- [47] M. Paasiavaara, S. Durasiewicz, and C. Lassenius, "Distributed agile development: Using scrum in a large project," in *2008 IEEE International Conference on Global Software Engineering*, Aug 2008, pp. 87–95.
- [48] R. Sindhgatta, B. Sengupta, and S. Datta, "Coping with distance: An empirical study of communication on the jazz platform," in *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion*, ser. OOPSLA '11. New York, NY, USA: ACM, 2011, pp. 155–162. [Online]. Available: <http://doi.acm.org/10.1145/2048147.2048190>
- [49] M. Hallikainen, "Experiences on agile seating, facilities and solutions: Multisite environment," in *2011 IEEE Sixth International Conference on Global Software Engineering*, Aug 2011, pp. 119–123.
- [50] P. Lous, M. Kuhrmann, and P. Tell, "Is scrum fit for global software engineering?" in *Proceedings of the 12th International Conference on Global Software Engineering*, ser. ICGSE '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 1–10. [Online]. Available: <https://doi.org/10.1109/ICGSE.2017.13>
- [51] K. H. Rolland, "Scaling across knowledge boundaries: A case study of a large-scale agile software development project," in *Proceedings of the*

Scientific Workshop Proceedings of XP2016, ser. XP '16 Workshops. New York, NY, USA: ACM, 2016, pp. 5:1–5:5. [Online]. Available: <http://doi.acm.org/10.1145/2962695.2962700>

- [52] F. O. Bjørnson and K. Vestues, “Knowledge sharing and process improvement in large-scale agile development,” in *Proceedings of the Scientific Workshop Proceedings of XP2016*, ser. XP '16 Workshops. New York, NY, USA: ACM, 2016, pp. 7:1–7:5. [Online]. Available: <http://doi.acm.org/10.1145/2962695.2962702>
- [53] I. Therrien and E. LeBel, “From anarchy to sustainable development: Scrum in less than ideal conditions,” in *2009 Agile Conference*, Aug 2009, pp. 289–294.
- [54] D. Wilby, “Roadmap transformation: From obstacle to catalyst,” in *2009 Agile Conference*, Aug 2009, pp. 229–234.
- [55] O. Ktata and G. Lévesque, “Agile development: Issues and avenues requiring a substantial enhancement of the business perspective in large projects,” in *Proceedings of the 2Nd Canadian Conference on Computer Science and Software Engineering*, ser. C3S2E '09. New York, NY, USA: ACM, 2009, pp. 59–66. [Online]. Available: <http://doi.acm.org/10.1145/1557626.1557636>
- [56] M. Rizwan and J. Qureshi, “Agile software development methodology for medium and large projects,” *IET Software*, vol. 6, no. 4, pp. 358–363, August 2012.
- [57] D. Talby and Y. Dubinsky, “Governance of an agile software project,” in *Proceedings of the 2009 ICSE Workshop on Software Development Governance*, ser. SDG '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 40–45. [Online]. Available: <http://dx.doi.org/10.1109/SDG.2009.5071336>
- [58] T. M. King, G. Nunez, D. Santiago, A. Cando, and C. Mack, “Legend: An agile dsl toolset for web acceptance testing,” in *Proceedings of the 2014 International Symposium on Software Testing and Analysis*, ser. ISSTA 2014. New York, NY, USA: ACM, 2014, pp. 409–412. [Online]. Available: <http://doi.acm.org/10.1145/2610384.2628048>
- [59] B. Sheth, “Scrum 911! using scrum to overhaul a support organization,” in *2009 Agile Conference*, Aug 2009, pp. 74–78.
- [60] V. Sithole and F. Solms, “Synchronized agile,” in *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*, ser. SAICSIT '16. New York, NY, USA: ACM, 2016, pp. 39:1–39:9. [Online]. Available: <http://doi.acm.org/10.1145/2987491.2987517>
- [61] M. Paasivaara, C. Lassenius, and V. T. Heikkilä, “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*, ser. ESEM '12. New York, NY, USA: ACM, 2012, pp. 235–238. [Online]. Available: <http://doi.acm.org/10.1145/2372251.2372294>
- [62] P. Rodríguez, K. Mikkonen, P. Kuvaja, M. Oivo, and J. Garbajosa, “Building lean thinking in a telecom software development organization: Strengths and challenges,” in *Proceedings of the 2013 International Conference on Software and System Process*, ser. ICSSP 2013. New York, NY, USA: ACM, 2013, pp. 98–107. [Online]. Available: <http://doi.acm.org/10.1145/2486046.2486064>
- [63] R. Vallon, C. Drger, A. Zapletal, and T. Grechenig, “Adapting to changes in a project’s dna: A descriptive case study on the effects of transforming agile single-site to distributed software development,” in *2014 Agile Conference*, July 2014, pp. 52–60.
- [64] R. Vivian, H. Tarmazdi, K. Falkner, N. Falkner, and C. Szabo, “The development of a dashboard tool for visualising online teamwork discussions,” in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 2, May 2015, pp. 380–388.
- [65] S. Hanly, L. Wai, L. Meadows, and R. Leaton, “Agile coaching in british telecom: making strawberry jam,” in *AGILE 2006 (AGILE'06)*, July 2006, pp. 9 pp.–202.
- [66] P. Mohagheghi and M. Jørgensen, “What contributes to the success of it projects?: Success factors, challenges and lessons learned from an empirical study of software projects in the norwegian public sector,” in *Proceedings of the 39th International Conference on Software Engineering Companion*, ser. ICSE-C '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 371–373. [Online]. Available: <https://doi.org/10.1109/ICSE-C.2017.146>
- [67] “The scrum guide,” <http://www.scrumguides.org/scrum-guide.html>, accessed: 2017-04-12.