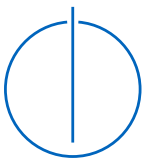
DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

Investigating the Current State of Research in Large-Scale Agile Software Development: A Systematic Mapping Study

Pascal M. Philipp



DEPARTMENT OF INFORMATICS

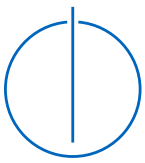
TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Information Systems

Investigating the Current State of Research in Large-Scale Agile Software Development: A Systematic Mapping Study

Untersuchung des aktuellen Forschungsstandes in der skalierten agilen Softwareentwicklung: Eine systematische Mapping-Studie

Author:	Pascal M. Philipp
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Ömer Uludağ
Submission Date:	16.12.2019



I confirm that this master's thesis in information systems is my own work and I have documented all sources and material used.

Munich, 16.12.2019 Pascal M. Philipp

Acknowledgments

First and foremost, I would like to thank my supervisor Ömer Uludağ for his support during this research project. I learned many important things from you and always enjoyed our close collaboration.

In addition, I would like to thank Professor Dr. Maria Paasivaara, Professor Dr. Casper Lassenius, and Professor Dr. Torgeir Dingsøyr for their valuable feedback.

Also, I would like to thank Professor Dr. Florian Matthes for the opportunity to write my thesis at his chair for Software Engineering for Business Information Systems (sebis).

Last but not least, I would like to thank my family and all my friends for their unconditional support during this exciting time.

Abstract

Contemporary organizations operate in an environment that is subject to constant change. Methods used in agile software development empower IT organizations to cope with the challenges of today's fast-paced business and technology environment. However, these development methodologies are designed for small, co-located, and self-organizing teams that produce software in close collaboration with business customers, utilizing regular feedback and rapid development iterations. The successful application of agile methods in small, co-located, and self-organizing teams inspired companies to use agile practices in large-scale projects. Hence, the first frameworks to support companies in scaling agile practices such as LeSS, SAFe, Spotify, and RAGE already exist and are increasingly used in large-scale agile settings. Simultaneously to the increasing popularity of large-scale agile development in industry, the popularity of the topic in science also increased. The studies published so far are mostly primary studies since there are only a few secondary studies on the topic. A continually increasing number of primary studies is an indicator that secondary studies can be conducted henceforth, to facilitate the aggregation of results of the primary studies. One type of secondary studies are Systematic Mapping Studies (SMSs) which are very well suited to give an overview of a field of research. Since within the field no SMS is conducted yet, this thesis aims to close this research gap by carrying out a SMS to reflect the current state of research activities in the field of large-scale agile development. The current state of the field will be determined in two phases. In the first phase, search engines will be searched, and the resulting studies are filtered to identify the relevant literature in the field. During the second phase, the selected studies from the first phase are used as input to perform a data extraction. During the systematic mapping, data is analyzed according to multiple dimensions. Finally, the key findings and promising future research directions are presented. The results show that the number of publications in the field of large-scale agile development has increased in recent years and that the field is gaining maturity because the number of publications in journals is rapidly growing. The most active research streams in the field are agile practices at scale and communication and coordination.

Contents

1. Introduction

This chapter contains the following sections. Section 1.1 describes the motivation of this master's thesis. Section 1.2 presents the objectives and the derived research questions. Following this, Section 1.3 describes the underlying research approach.

1.1. Introduction

> Indeed, one of my major complaints about the computer field is that whereas Newton could say, "If I have seen a little farther than others, it is because I have stood on the shoulders of giants," I am forced to say, "Today we stand on each other's feet." Perhaps the central problem we face in all of computer science is how we are to get to the situation where we build on top of the work of others rather than redoing so much of it in a trivially different way. Science is supposed to be cumulative, not almost endless duplication of the same kind of things.
>
> *Richard Hamming*
> *1968 Turning Award Lecture*

Nowadays, organizations operate in an environment that is subject to constant economic, political, and technological change; an environment in which flexibility, customization, and learning are watchwords and adaptability is the key to success [1]. Methods used in agile software development such as Extreme Programming, Lean Development, Scrum, and others empower large IT organizations to cope with the challenges of today's fast-paced business and technology environment [2]. These development methodologies are designed for small, co-located, and self-organizing teams that produce software in close collaboration with business customers, utilizing regular feedback and rapid development iterations [3]. The successful application of agile methods in small, co-located, and self-organizing teams inspired companies to use agile practices in large-scale projects [4]. The first frameworks to support companies in scaling agile practices such as LeSS, SAFe, Spotify, and RAGE already exist [5, 6] and are increasingly used in large-scale agile settings [7]. Simultaneously to the increasing popularity of large-scale agile development in industry, the popularity of the topic in science also increased [8, 9]. As a result, the number of publications that increase knowledge in the field of

large-scale agile development has been steadily rising [8, 9]. According to Moe and Dingsøyr [9], most publications made a scientific contribution in areas Coordination [10, 11], Knowledge sharing [12, 13], and Portfolio management [14]. Regarding the use and tailoring of the scaling frameworks [15], and for the tailoring of other methods at scale [10, 16, 17, 18, 19, 20] less scientific work was published. The publications published so far are mostly primary studies, i.e., research projects that collect original data, since there are only a few secondary studies on the topic, such as the structured literature reviews by Uludağ et al. [21] and Dikert et al. [15]. A continually increasing number of primary studies is an indicator for the maturity of a research area and leads to a critical tipping point, which makes Bernard of Chartres statement "standing on the shoulders of giants" a reality for the respective research area, since secondary studies can be conducted henceforth, which facilitates the aggregation of the results of the primary studies. The aggregation of primary studies can, for example, take form of an overview or summaries of the individual publications [22]. Systematic Literature Reviews (SLRs) are an instance of secondary studies and have been increasingly used in recent years [22, 23]. The enormous popularity of SLRs goes hand in hand with the increasing popularity of evidence-based research paradigm in software engineering, which promotes the use of empirical and systematic research methods and originally stems from medical research [22]. Kitchenham et al. [23] investigated the adoption possibilities of SLRs used in medicine and presented them in conjunction with guidelines to perform SLRs in the software development domain. A significant advantage of SLRs over classical literature reviews is the clearly defined SLR methodology, which aims to minimize bias [22]. Systematic Mapping Studies (SMSs) are similar to SLRs because they take advantage of many elements of the SLR methodology. SMS differ from SLRs in their reduced complexity, as they examine broader research questions which do not require in-depth analysis but high-level analysis [24]. In contrast to SLRs, SMSs are very well suited to give an overview of a field of research [22]. In the field of large-scale agile development, there is no SMS published yet, although the available number of primary studies would be sufficient because, as mentioned above, some SLRs have already been carried out. This work aims to close this research gap and to carry out an SMS to provide an overview of the research activities in the field of large-scale agile development.

1.2. Research objectives

This thesis aims to present the current state of research in the field of large-scale agile development. Therefore, twelve research questions are defined. Each of these research questions intends to investigate the publication landscape according to a particular

characteristic. Hereafter, this section describes each research question. The work of Rodríguez et al. [25] and Haghighatkhah et al. [26] inspired the design of the RQs of this thesis. It is noteworthy that their RQs are similar to the RQs in this thesis, but were designed for a different context.

> ***Research question 1:*** *What is the distribution of publications per year in the area of large-scale agile development?*

To answer the first research question, the publication year of each study is determined, and then the number of studies published in the same year is counted. In addition to the information about the number of studies in a year, the cumulative sum is calculated. The information is presented with the help of a frequency line chart.

> ***Research question 2:*** *How are studies in the large-scale agile development research geographically distributed?*

To answer the second research question, the country in which the research institution of the first author is located is determined for each study. Subsequently, it is counted how often a single country appears in the publication landscape. This information is illustrated using a map chart and a frequency bar chart.

> ***Research question 3:*** *Which studies are the most influential in terms of number of citations?*

To answer the third research question, the number of citations of each study is determined with the help of Google Scholar. This information is displayed with a bar chart which shows the most salient studies in the field.

> ***Research question 4:*** *Who are the most salient scientists in the field of large-scale agile development measured by number of publications and citations?*

To answer the fourth research question, the total number of citations and studies for each author is determined. This information is displayed with the help of two bar charts, which provide a quick overview of the most salient authors in the field.

> ***Research question 5:*** *What are publication channels of studies related to large-scale agile development?*

To answer the fifth research question, the publication channels of the studies are determined. The determination of the publication channels comprises, on the one hand, the determination of the publication venue name and on the other hand, the identification of the publication channel type. Studies will be assigned to one of the channel types journal, conference, or workshop. Subsequently, the number of studies published in a channel of the same name or type will be counted. The results are presented using two frequency bar charts.

Research question 6: *What research types of the studies have been used in large-scale agile development?*

To answer the sixth research question, the research types of studies are determined. For this purpose, the number of studies for the three research types are determined. Examples of types are evaluation research, philosophical studies, and solution proposals. A description of the research types which were identified during the data extraction phase is given in Table 4.9. The results are visualized with the help of a pie chart.

Research question 7: *What research approaches have been used in studies related to large-scale agile development?*

To answer the seventh research question, the research approaches of the studies in the field of large-scale agile development are determined and the studies of a certain approach are counted. The research approaches are categorized as survey, design and creation, case study, action research, grounded theory, mixed methods, SLR/SMS, and theoretical studies. All studies that cannot be allocated to any of these categories are labeled as not applicable. A description of the research types which were identified during the data extraction phase is presented in Table 4.10. The results are presented with the help of a frequency bar chart.

Research question 8: *What kinds of contributions are provided by studies related to large-scale agile development?*

To answer the eighth research question, at first, the research contribution of each study is determined. Then the count of studies with the same contribution type is identified. The research contributions can be divided into the categories model, theory, framework or methods, lessons learned, and guidelines. A description of the research contributions identified during the data extraction phase is given in Table 4.11.

Research question 9: *What is the proportion of primary and secondary research?*

To answer the ninth research question, the number of studies representing primary or secondary research is determined. The results are presented with the help of a pie chart.

Research question 10: *What is the rigor and relevance of the studies in large-scale agile development?*

To answer the tenth research question, a score for the rigor and a score for the relevance is calculated for every study. The rigor of a study is evaluated by the aspects context, study design, and validity (Table 4.13). Likewise, the assessment of the relevance of a study is based on aspects. The four aspects of relevance are subjects, context, scale, and research method (Table 4.12). The results are presented in a bubble chart.

Research question 11: *Which research streams exist in large-scale agile development and which are the most active ones?*

To answer the eleventh research question, the topic of each study is determined. During an iterative process, the identified set of topics is summarized, and subtopics are created. The results are arranged in a tree graph (see Figure 5.24) in which nodes represent the research streams of large-scale agile development. The branches (several nodes connected by edges represent a branch) of the tree structure the field in topics and subtopics.

Research question 12: *What are promising future research directions in the area of large-scale agile development?*

To answer the twelfth research question, the future work sections of the studies are searched for promising research ideas. Similar to research question 11, the results are presented in a graph tree. The identified research ideas are assigned to the topics in large-scale agile development.

1.3. Research approach

In this thesis, a SMS was chosen as a research approach to answer the RQs defined in the previous Section 1.2. The SMS approach can provide an answer for the RQs because it creates a structure of the type of research reports and results through categorization. SMSs are also suitable for visually summarizing the results by in a map [22]. Therefore, in this work a visual representation of the topics and promising research ideas within the field was implemented to help scientists navigate their future research efforts. Hereafter, the SMS approach applied to the context of this thesis is described, which follows the suggestions of Kitchenham et al. [23] and Petersen et al. [22]. On a higher level, the SMS approach can be understood as a process consisting of two phases (see Figure 1.1). In the first phase, two automated searches within six search engines, and a manual search was conducted to identify studies in the field of large-scale agile development. Then the identified studies underwent a filtering process where their relevance was assessed. At the end of the first phase, 108 studies were assessed as relevant. This set of relevant studies represented the input of the second phase, in which data extraction and systematic mapping were carried out. Below, both phases are described in more detail, and a visual overview is presented in Figure 1.2.

Figure 1.1.: A high-level overview of the research approach

1.3.1. Identification and filtering of the of sources

The first phase includes the identification of high-quality publications relevant to large-scale agile development research. For a systematic implementation of this phase, research questions and a search strategy were defined. The search strategy is inspired by Zhang et al. [27]. In their paper, they propose practical guidelines for the implementation of a search strategy, and therefore compensate what the guidelines for SLRs lack [27]. Zhang et al. [27] conclude that implementing their suggestions will result in both an effective search strategy design and search strategy execution. Subsequently, inclusion and exclusion criteria were derived from the research questions, which, together with the search strategy, influenced the definition of the search terms of the preliminary search (PS) and main search (MS). Within the PS, a manual and an automated search in six search engines (see Table 4.1) were implemented. The execution of a manual search had several objectives. The first goal was to ensure that all relevant publications, which were already known to the scientists in advance, would not be lost. The second goal was to increase the quality of relevant studies resulting from the preliminary search, as this set was used to assess the effectiveness of the search string belonging to the automated search. After removing the duplicates within the PS and MS, the results of both searches were merged. Thereafter, duplicates between both searches were removed. The filtering step was performed by two scientists (see Figure 4.3). Within the process the filter levels duplicate, full-text-access, meta-data, abstract, full-text, final review, and final decision were introduced to increase the efficiency since for each filter level the effort to check the relevance of a study within it (e.g., abstract analysis) was lower than the effort for an assessment in the following level (e.g., full-text analysis). The 108 selected studies are listed in Appendix A.1.

1.3.2. Data extraction and systematic mapping

The steps of the data extraction process were performed multiple times to create a systematic map for thirteen categories (see Figure 4.4). The thirteen categories (C1-C13) are described in Table 4.8. The input for each iteration of the process were the 108 studies (see Appendix A.1), which were identified as relevant during the filtering process. Similarly to the filter process, both scientists performed the data extraction process separately to prevent mutual influence. The first step in the process, the keywording, was carried out if the categories could not be classified directly, or no classification scheme could be found in the literature. As a result, no classification schema was required for the categories C5, C6, C7, and C13. During the keywording, each scientist selected the words, which describe the properties of a study for the respective category best. As mentioned above, both scientists performed this step

separately. After both scientists finished the keywording step, conflicts between the scientist were resolved. Then, both scientists worked separately with the final keywords to derive a classification scheme. Again, deviations between the two classification schemes were resolved. In the last step, each scientist assigned the studies to the final classification scheme. After all assignment conflicts were resolved, the mapping was final. Similar to the filtering process, an Excel spreadsheet was used for the data extraction process in which the columns of the inactive scientist were collapsed to avoid psychological bias. In the following, the classification schemes of the categories are described.

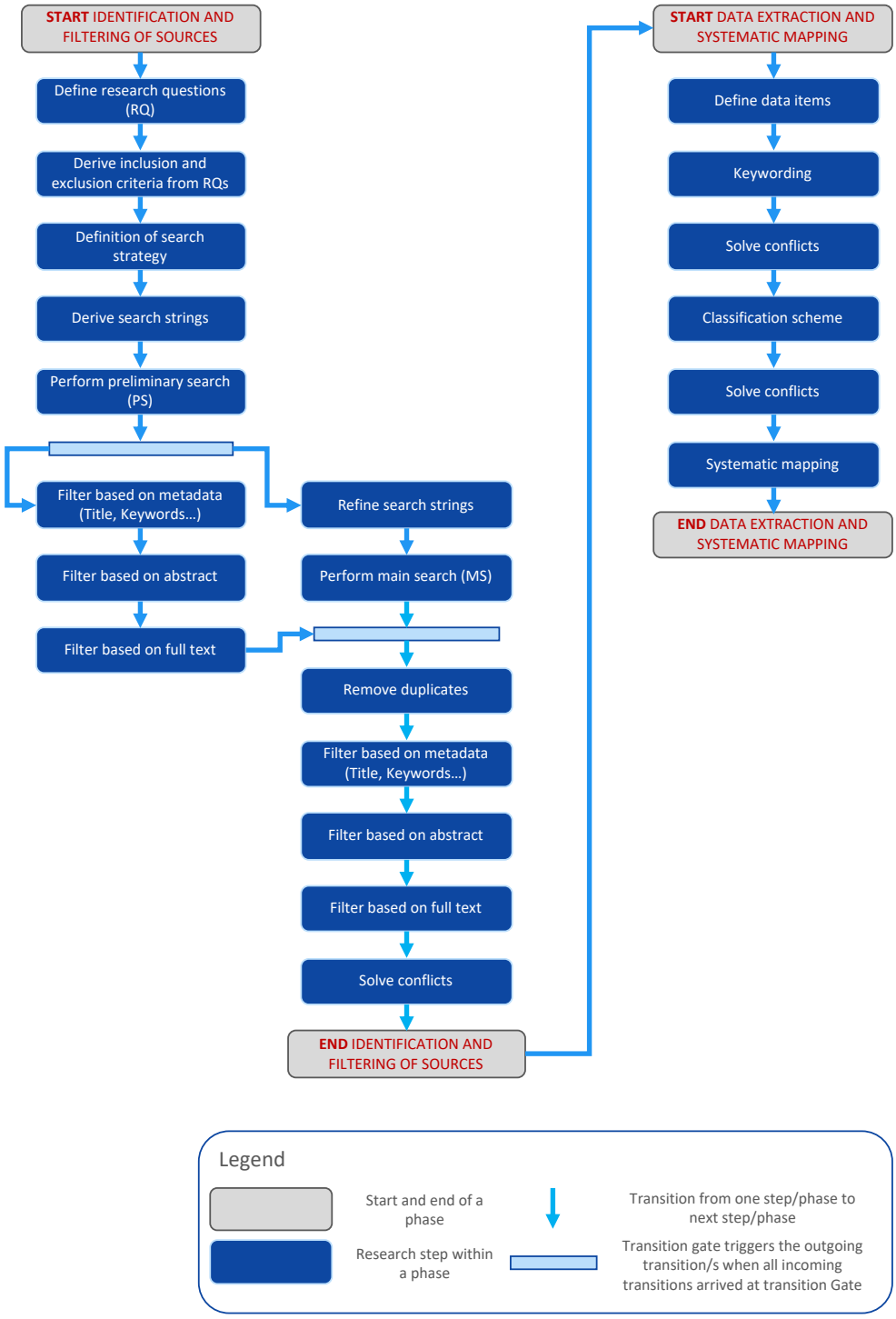


Figure 1.2.: Detailed overview of the research approach

2. Foundations

This chapter is dedicated to providing the theoretical foundations for the remaining chapters of this thesis. Therefore, it aims at identifying relevant research findings and defining domain-specific terms as well as key concepts.

2.1. Agile software development

This section aims at creating a shared understanding of agile software development. Therefore, at first, the agile manifesto is described by defining agility, and by presenting the principles and values behind it. Subsequently, the Scrum roles, artifacts, and the Scrum process is described. Finally, Kanban and its relation to Scrum is discussed.

2.1.1. Manifesto for agile software development

In February 2001, seventeen people came together to find an alternative to the traditional documentation driven and heavyweight software development processes [28]. They chose the word agile to describe the characteristics of this alternative approach [29] and created the Manifesto for Agile Software Development, which was signed by all participants to demonstrate the consensus on the principles and values it contains [28]. The recognized experts in the field of global competitiveness Nagel et al. [30] describe agility as follows:

> "Agility is dynamic, context-specific, aggressively change-embracing, and growth-oriented. It is not about improving efficiency, cutting costs, or battening down the business hatches to ride out fearsome competitive 'storms.' It is about succeeding and about winning: about succeeding in emerging competitive arenas, and about winning profits, market share, and customers in the very center of the competitive storms many companies now fear."

When agility is examined in the context of software development, it becomes apparent that, firstly, light-but-sufficient rules of project behavior and, secondly, the use of human- and communication-oriented rules form the main characteristics [29]. In addition to the twelve principles (see Table 2.1), four overarching values emerge from the agile

manifesto, which should also contribute to increased agility within software projects. In the following, the four values are listed and extended by an interpretation by Stellman and Greene [31]:

- "Individuals and interactions over processes and tools" means that it is more important for agile teams to focus on the team members and their communication than to pay attention to the tools and practices used.
- "Working software over comprehensive documentation" states that software which satisfies a user's need is more valuable than a specification describing the need.
- "Customer collaboration over contract negotiation" means that from now on, customers will be seen as team members.
- "Responding to change over following a plan" means that plans rarely come true and that it is, therefore, more important to deliver software than to work on a plan.

Please be advised that understanding that although all items on the left side are deliberately given more importance, the items on the right side should not be avoided entirely, as they can also lead to benefits [28]. A generally accepted fact is that agile methods are in widespread use and that projects using traditional methods have difficulty responding to changes in objectives, resources, materials, techniques, and tools [32, 33, 34]. Further, please note that agile methods cannot be used effectively without any pre-planning, because for example, a significant amount of work is performed before the project starts because the identification of the requirements of the first release necessitates intensive communication with the customers [34, 35].

2.1.2. Scrum process framework

> One of the charms of the Rugby Union game is the infinite variety of its possible tactics. [...] with the ball in its hands, a team is in a position to dictate tactics which will make the best use of its own particular talents, at the same time probing for and exposing weaknesses in the opposing team. The ideal team [...] will make sure that the possession won by the forwards is employed to the maximum embarrassment of the opposing team.
>
> *Takeuchi and Nonaka*
> *The new product development game [37]*

Scrums co-creators Ken Schwaber and Jeff Sutherland [38] define Scrum as a framework that helps people solve complex problems while delivering products with the highest

Principle	Quotation
P1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
P2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
P3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
P4	Business people and developers must work together daily throughout the project.
P5	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
P6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
P7	Working software is the primary measure of progress.
P8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
P9	Continuous attention to technical excellence and good design enhances agility.
P10	Simplicity–the art of maximizing the amount of work not done–is essential.
P11	The best architectures, requirements, and designs emerge from self-organizing teams.
P12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Table 2.1.: Principles behind the Agile Manifesto by Beck et al. [36]

possible value productively and creatively. Inspired by the groundbreaking work of Takeuchi and Nonaka [37], Scrum was named after the Scrum formed in rugby sports [39]. Please note that Scrum is not just a technique, process, or a final method, but a flexible framework that allows the simultaneous use of various processes and techniques [38]. Scrum consists of the three components roles, process, and artifacts [40], which are all in constant interaction with the Scrum teams [38]. These components are described in detail below.

Roles

The responsibilities of a project are assigned in Scrum to the three roles Product Owner, Team, and Scrum Master [41]. An important feature of Scrum is that each team member can autonomously decide how to solve the assigned tasks to ensure that nobody can interfere with the Scrum process [41, 42]. All three Scrum roles are described below. The Product Owner (PO) of a Scrum project identifies the requirements of the product in close and continuous cooperation with customers, users, and other stakeholders and

subsequently communicates these to the team, which implements the requirements [43]. The PO is, therefore, responsible for project effectiveness and, consequently, for ensuring that the product meets customer requirements [43]. The main task of the Scrum team is to implement the requirements defined by the PO [41]. Scrum teams share the following characteristics: they are cross-functional, self-organizing, and consist of five to ten members [40]. The Scrum Master (SM) is responsible for establishing the Scrum practices and values and removes obstacles to enable a smooth Scrum process [40, 43]. Thus, the SM fulfills a support function for the process [44] and is responsible for the efficiency of the project.

Artifacts

Scrum consists of the three artifacts product backlog, sprint backlog, and burndown charts. Both the product backlog and the sprint backlog are used to record, allocate, and prioritize the requirements of the agile project [41]. The third artifact, the burndown chart, is a tool that makes it possible to estimate, based on the requirements, how much time a project will take until the product meets all customer expectations [41]. All artifacts are described in more detail below. The product backlog is a list that contains all requirements of the project and is continuously evolving as it adapts to the changes the project undergoes [41]. If the project loses funding, the product backlog will no longer be used [41]. The PO has the responsibility to guarantee the availability of the product backlog, to prioritize it, and to ensure the quality of all its entries [41]. The product backlog may only be changed during the project kickoff and subsequent sprint planning meetings, which are recurring in sync with the project cycle, and is locked during all other project phases [40]. If the two backlogs are understood as a set of requirements, then the sprint backlog is a subset of the product backlog, which contains precisely as many requirements as the team is confident to complete in the next sprint [40]. The fact that the number of requirements corresponds to the team capacities is ensured by an essential characteristic of Scrum, which mandates that only the team decides on the requirements transferred from the product backlog into the sprint backlog [40, 43]. The last artifact in Scrum is the burndown chart, which represents the time remaining until the end of the project [41]. An example project burndown chart is shown in Figure 2.1. The vertical axis measures the remaining project days, which should decrease as the project progresses. This reduction is called burn down. With the help of historical data, a trend line can be drawn (in the illustration, the trend line is represented by the blue line), which makes it possible to estimate the remaining days until the end of the project.

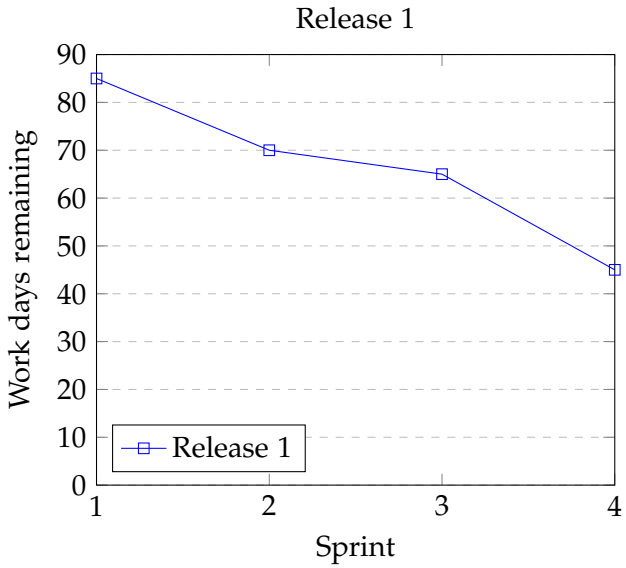


Figure 2.1.: Burndown chart by Schwaber [41]

Process

To deliver products of the highest possible value, Scrum follows a process on which all of its practices are build on [41], as illustrated in Figure 2.2. All Scrum practices are part of a process that allows the delivery of products in the best possible quality [41]. In every Scrum project, there is a vision about the system that will be developed during the project [41]. Firstly, the PO must derive from this vision all the requirements necessary to implement the vision; secondly, the PO must ensure that the implementation is designed in such a way that the return on investment of the investors financing the project is maximized [41]. The basic building block of Scrum is the iterative and incremental process that is used to implement the requirements of the product backlog [41]. Each iteration begins after completion of the Sprint Planning Meeting, in which the requirements with the highest priority in the product backlog are identified in close cooperation between the PO and the team members and transferred to the sprint backlog [41]. During a sprint, the team works through the requirements of the sprint backlog and meets every day to hold a meeting called Daily Scrum meeting to synchronize the work of the team members [41]. After the sprint review meeting, the SM encourages all team members to participate in an event called Scrum retrospective, which is the last event of the iteration and aims to have the team reflect on the past sprint in order to identify actions that will enable an effective and enjoyable next sprint [41]. Furthermore, the Scrum retrospective starts the next iteration because it is followed

by the sprint planning meeting of the next sprint [41].

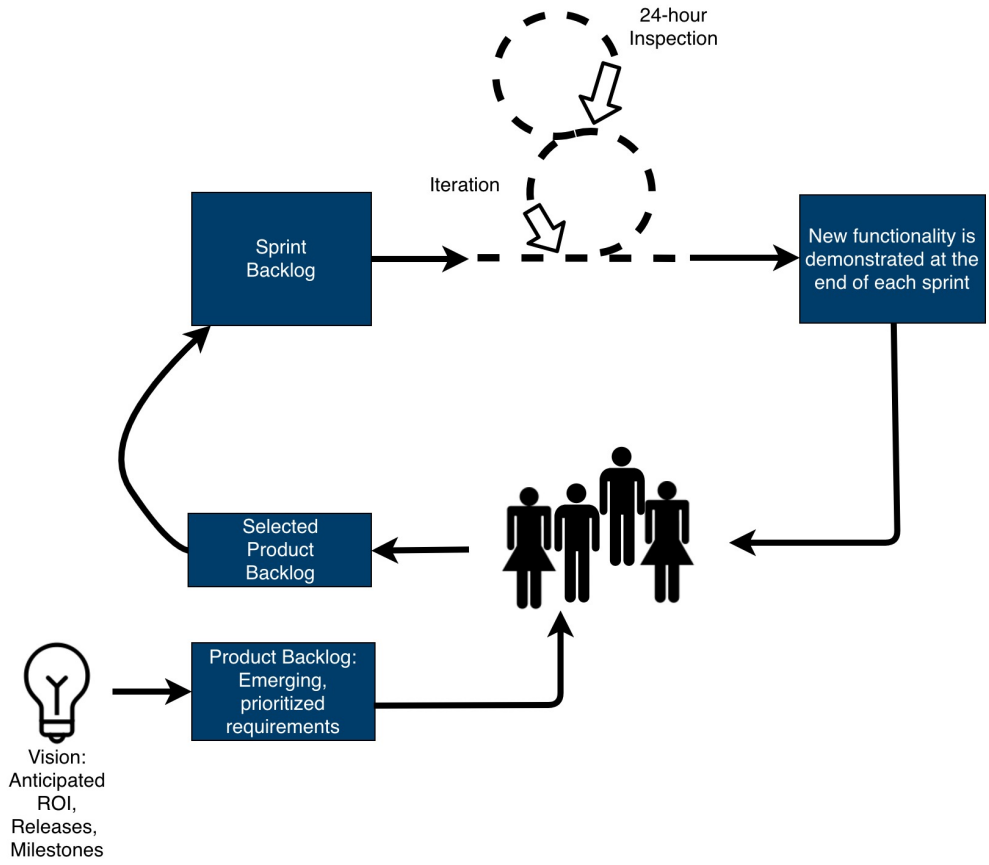


Figure 2.2.: Scrum process overview by Schwaber [41]

2.1.3. Kanban in software development

Kanban was first successfully applied in the 1950s at Toyota, a company active in the Japanese manufacturing industry [45, 46]. Decades later, in 2004, Kanban was introduced into the software development industry at Microsoft by David J. Anderson, who, with the help of Kanban, guided a poorly performing software team back on a successful course [45, 46]. While Kanban gained importance in the software industry, the first attempts were made to combine Kanban with Scrum, which is a very established method of agile software development [47]. Whereas Scrum follows an incremental and iterative process to deliver products with the highest possible value, Kanban focuses on optimizing the workflow, minimizing the average development time of a work

item, and limiting the work in progress (WIP) [38, 48]. A hybrid, which consists of elements of Kanban and Scrum, is often called Scrumban in the literature [49, 50, 51, 52, 53]. Scrumban's goal is to support companies that are exposed to constantly changing customer requirements and recurring coding problems [49]. In the following, Kanban is described in more detail. Kanban is a visualization of a pull system for software development projects, which is used to manage the workflow in a project [54]. Managing the workflow is done firstly by setting a WIP limit, and secondly, by identifying constraints and finally by coordinating work within the team [54]. The limit for the WIP is set based on the team's working capacity, which allows a balance to be maintained between demand and team throughput [46]. This balance fosters a sustainable development rhythm, which can lead to improved team performance on the one hand and improve the quality of the developed products on the other [46]. The implementation of Kanban in a software project is done with the help of a Kanban board, which is the visual representation of a value stream [55]. This value stream is divided into columns that represent the current status of the work and through which the work items flow [55]. However, each of these columns is only allowed to contain a pre-defined number of items, which ensures that the WIP is limited [55]. An item leaves one column and moves to the next as soon as pre-defined requirements are met, which specify the state an item must be in in order to be considered done [55]. Ultimately, the wandering of items through their work states can be visually perceived as a value stream [55].

2.2. Large-scale agile development

> My dear, here we must run as fast as we can, just to stay in place. And if you wish to go anywhere you must run twice as fast as that.
>
> *Lewis Carroll*
> *Alice in Wonderland*

This section aims to create a shared understanding of large-scale agile development. First, a taxonomy is presented to define the term large-scale. Then factors are presented that require agility to be scaled. Finally, an overview of the most common scaling agile frameworks is given, and success factors and challenges for scaling agile practices are presented.

2.2.1. Taxonomy

The success of agile methods in projects that can be realized by small and co-located teams inspired their use in other domains as well as their application in large-scale

Level	Number of teams	Coordination approaches
Small-scale	1	Coordinating the team can be done using agile practices such as daily meetings, common planning, review and retrospective meetings.
Large-scale	2-9	Coordination of teams can be achieved in a new forum such as a Scrum of Scrums forum.
Very large-scale	10+	Several forums are needed for coordination, such as multiple Scrum of Scrums.

Table 2.2.: A taxonomy of scale of agile software development projects by Dingsøyr et al. [56]

projects [19]. Large-scale projects are complex and require a large number of people for their implementation [8]. They are often of great importance for companies but also for entire countries [8]. Despite the increasing popularity of the use of agile methods in large-scale projects, the term large-scale is not clearly defined [8, 56]. To address this problem, Dingsøyr et al. [56] developed a taxonomy of scale for the classification of agile software development projects. This taxonomy is shown in Table 2.2 and assigns not only the scale to an agile software development project based on the number of teams but also the appropriate coordination approach for the project.

2.2.2. Scaling factors

In addition to team size, Ambler and Lines [57] present five other factors (see Figure 2.3) that, when magnified, can lead to a strong need for scaling agile practices [58]. Each factor is briefly explained hereafter. Teams can be small with two members or very large with more than 200 members [59]. Large teams are needed to implement projects with a high domain or technical complexity [59]. The next scaling factor is called geographical distribution and gives information about the distance between the teams. [59]. For example, the teams can be co-located if they work on the same floor in the same building, or they even work in different time zones, thus causing a high scaling need [59]. If the organizational factor is significant, teams from different organizations will work together on a project, whereas if the organizational factor is weak, the teams will work in the same division [59]. Concerning compliance, externally imposed compliance requirements, in particular, are scaling drivers [59]. A high compliance factor results, for example, from life-critical regulations [59]. The low domain complexity is present, for example, in the development of a website that only displays information [59]. An e-commerce website would lead to greater domain complexity due to the higher implementation effort [59]. The technical complexity, for instance, is low in the development of brand-new stand-alone applications using new technologies [59]. If existing solutions and data sources are transformed into a new

solution, the technical complexity increases [59].

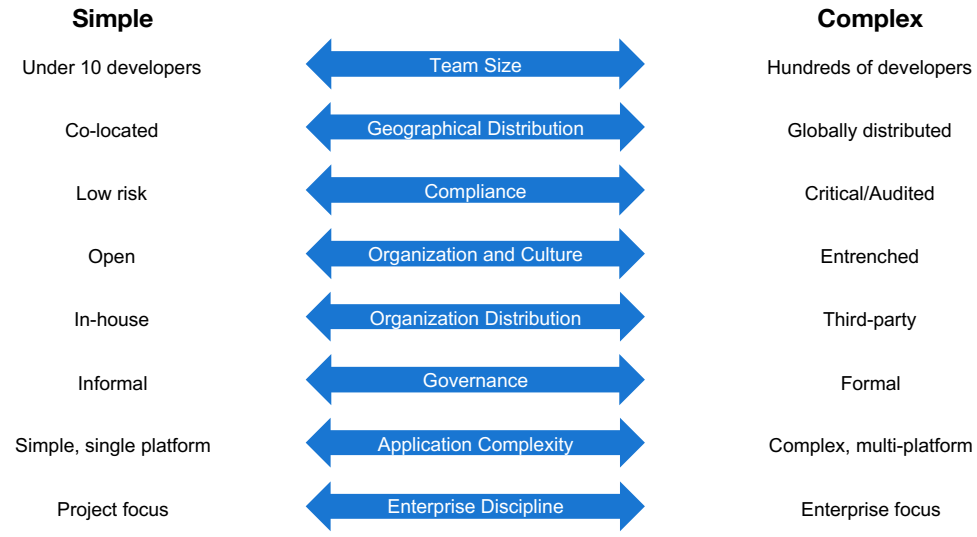


Figure 2.3.: Scaling factors affecting IT teams by Ambler and Lines [57]

These six scaling factors are part of the Software Development Context Framework (SDCF), which is based on the work of Kruchten [58, 60]. Kruchten's [60] contextual model for the development of software-intensive systems helps to adopt and adapt agile practices. This model is intended to bring the most significant benefits to agile projects located outside the so-called sweet spot. This sweet spot is illustrated in Table 2.3. For each of the eight criteria, a benchmark is presented to define the sweet spot (blue font).

2.2.3. Scaling agile frameworks

Uludağ et al. [21] identified twenty scaling agile frameworks (see Table 4.5) with the help of a structured literature review. Most frameworks are based on traditional agile methods such as XP and Scrum and have been adapted to new requirements to apply to massive projects where multiple teams work. Also, Uludağ et al. [21] calculated maturity scores for the frameworks. The three frameworks with the highest maturity scores are Large Scale Scrum, Scaled Agile Framework, and Disciplined Agile 2.0.

Criteria	Benchmark
System Size	0 ..12 ...300
Criticality	Simple, $ losses, ...deaths
System Age	Exploratory, greenfield, legacy maintenance
Rate of Change	Low, medium, high
Business Model	In house, Open Source, ...
Stable Architecture	Stable, changed, new
Team Distribution	Colocated, ..., ..., offshore, outsource
Governance	Simple rules, ..., SOX, ...

Table 2.3.: A particular context: the Agile sweet spot (in blue) by Kruchten [61]

3. Related Work

This chapter identifies and summarizes the secondary research relevant to large-scale agile development. The presented related work further extends on the foundations presented in the last chapter.

3.1. SLRs on large-scale agile development

Arif et al. (2015)

Arif et al. [62] conducted an SLR to present a detailed analysis of the limitations of agile methods in large-scale agile projects. Also, the SLR served to examine the existing literature for recommendations for action and to examine their limitations. According to the authors, the obstacles identified for scaling agile methods include problems with documentation, time period, human resources, and budget overflow issues [62]. Please note that it was not explicitly stated that the research process was conducted according to recommendations proven in the past. Kitchenham et al. [63] were listed in the bibliography, but there is no quote of this source within the paper.

Abrar et al. (2015)

Abrar et al. [64] conducted an SLR to identify motivators of managers for introducing agile methods in large-scale projects. In total, they identified 21 motivators. Some of the motivators were classified as eminently important because they are critical of all pre-defined variables, such as company size [64]. These motivators include strong management support for the scaling project, agile development environment training and learning, existing expertise in agile development methods, high team competence, and top management briefing to create a better understanding of agile [64]. The motivators were derived from 58 papers which could be identified with the help of the SLR. The research methodology was conducted according to the guidelines for SLRs developed by Kitchenham et al. [63, 65]. Accordingly, the research process was divided into three sub-steps: drafting a review plan, review execution, and review reporting.

Alsaqaf et al. (2017)

Alsaqaf et al. [66] used a SLR to identify 12 challenges that can affect quality requirements within agile projects. Quality requirements are the requirements that describe the qualities of the system [66]. Examples of system qualities are performance requirements and maintainability requirements [66]. Furthermore, they evaluated 13 proposals for the correct handling of quality requirements in agile projects. The research methodology was based on the guidelines for SLRs, according to Kitchenham et al. [63, 65, 66]. The relevant papers were identified with the help of pre-defined inclusion and exclusion criteria. Subsequently, the data of the relevant paper were extracted with the help of a thematic synthesis [66]. Based on the extracted data, an inductive coding, according to Saldana's open coding approach was performed to answer the research questions [66, 67].

Dikert et al. (2017)

With the help of a SLR of 52 papers, Dikert et al. [15] identified challenges and success factors for large-scale agile transformations. Please note that most of the papers in this research project were experience reports (46), and only a few academic studies (6) could be identified [15]. The 35 identified challenges were divided into nine categories, and the 29 identified success factors were divided into 11 categories [15]. The most critical challenges include the integration of other functions, the difficulty of implementing, and change resistance [15]. The most crucial success factors include choosing and tailoring, management support, and mindset [15]. The research methodology is based on the recommendations for conducting SLRs in software development research developed by Kitchenham et al. [63, 65]. Dikert et al. [15] implemented Kitchenham et al. [63, 65] methodology by dividing the research process into four sub-steps: identification of potential sources, filtering of relevant sources, coding as well as analysis of coding and aggregation. With the help of previously defined inclusion and exclusion criteria, 52 relevant papers were filtered out as a result of the first two steps [15]. These were used in the following steps to identify the challenges and success factors mentioned above [15].

Uludağ et al. (2017)

Uludağ et al. [21] used a structured literature review to describe architectural roles that occur within Scaling Agile frameworks. The research project identified 20 different frameworks and analyzed three of those in more detail, including SAFe. In the following, findings regarding SAFe concerning the architecture role software architect are described in more detail. The role of the Enterprise Architect aims to successfully

integrate technology used within individual Value Streams into the system landscape of all Value Streams, requiring close collaboration with the System and Solution Architects and business stakeholders [21]. An essential task of enterprise architects is to define the enterprise strategy and communicate it to the system architects and their non-technical stakeholders [21]. The role of the software architect in SAFe is fulfilled by the System Architect that can be occupied by either a single person or a team and aims to make architectural decisions for the associated Agile Release Train (ART) so that they are consistent with the architectural strategy for the overall solution [21]. The software architect focuses mainly on the program level within SAFe through active support of the ART teams [21]. His primary responsibilities include defining subsystems and related interfaces as well as non-functional-requirements, developing and presenting an architectural vision, and supporting the PO [21]. The role of the Solution Architect in SAFe is occupied by cross-functional teams, which have an understanding of the entire system landscape of a solution and resemble the System Architect in terms of responsibilities, with the difference that the main focus is on the Value Stream Level [21].

3.2. SMSs on state of the art in agile software development

Freitas da Silva et al. (2011)

Freitas da Silva et al. [68] carried out an SMS to investigate the state of the art of agile software product lines (SPL). The authors describe SPL as a systematic approach to reusing commonly used artifacts that improves product delivery. According to the authors, finding the right balance between SPL and agile development activities can lead to benefits in dealing with changing customer requirements, managing product variability, and reusing software artifacts [68]. Their research project enabled them to identify gaps in the current research and to aggregate the current evidence [68]. Furthermore, with their SMS, they provide a better understanding of the integration of Agile and SPL [68]. They based their research methodology on the recommendations of Peterson et al. [22] and combined them with recommendations for SLRs, such as the implementation of a protocol definition. The identification of the paper was carried out with the help of a manual and an automatic search [68]. The papers found were then classified as relevant or irrelevant using inclusion and exclusion criteria. Finally, a reference search was conducted to find other relevant papers.

Magües et al. (2016)

Magües et al. [69] set themselves the goal with their SMS to determine the current state of the research regarding the integration between agile processes and usability. In agile software development, usability is implemented with the help of user-centered design (UCD). According to the authors, there has been a strong focus in science on the combination of agile software development methods and UCD [69] . Accordingly, the authors argue for the implementation of an SMS, since this research method both gives an overview of the research field and uncovers possible research gaps [69] . The research methodology was designed according to the recommendations of Peterson et al. [22]. In the selection phase, papers were checked for relevance by assessing the titles first. Afterward, the abstract, introduction, and conclusion were assessed. The selection process was applied both in a preliminary approximation search and subsequently in a second search. Within the second search, a reference search was performed in contrast to the approximation search. In the conclusion of the paper, the authors mention that the papers identified as relevant can be assigned to the four different perspectives: processes, practices, team, and technology [69]. Furthermore, no systematic approach for the integration of human-computer interaction activities and techniques with agile processes could be found with the help of SMS, and thus this area could be identified as a research gap [69].

Arias et al. (2016)

Arias et al. [70] carried out an SMS with a focus on the application of process mining in agile software development. Within this context, process mining aims at deriving improvement suggestions for the future project progression based on historical log data collected during a project. With the SMS they aspired to reflect on the one hand the current state of scientific publications, which addressed the application of process minings in agile software development [70]. Furthermore, they set themselves the goal of deriving characteristics from the set of identified studies with their SMS [70]. The authors designed their research methodology based on Kitchenham et al. [71] guidelines for the implementation of SLRs, which also contain recommendations for the implementation of SMSs. After defining the research questions, inclusion and exclusion criteria were used to identify relevant papers [70]. For each relevant paper, a reference search was carried out to extend the selection process and to minimize the possible bias [70].

Curcio et al. (2018)

Curcio et al. [72] present with their SMS an overview of science in the field of requirements engineering in the context of agile software development. In advance, they motivate their research projects with the fact that agile projects often fail due to errors in requirement engineering [72, 73]. The research approach is based on the recommendations for SMS by Kitchenham et al. [63, 65] and Peterson et al. [22]. The search strategy includes the definition of the search space, the inclusion and exclusion criteria, a search string as well as a reference search. Be advised that in addition to the recommendations for SMS by Kitchenham et al. [63, 65] and Peterson et al. [22], the authors conducted a quality assessment following the recommendations by Dybå and Dingsøyr to ensure that all filtered studies are relevant for answering the research questions [74].

3.3. SMSs in agile software development

Yang et al. (2016)

Yang et al. [75] conducted an SMS to not only determine the current state of scientific literature on the combination of agile methods and architecture but also to synthesize data from the identified studies. The authors justify their research project by identifying the topic as one of the most critical research questions in their field [75, 76]. The research methodology is based on the recommendations of Peterson et al. [22]. Before the main search was conducted, the authors conducted a trial search to optimize their search strategy and search strings. The primary studies found during the main search underwent three rounds of selection. In the first selection round, the relevance of the paper was evaluated with the help of the title. All papers evaluated as relevant were then assessed in the second selection round based on the abstract. In the last selection round, a full-text analysis was performed.

Ramírez-Mora et al. (2017)

Ramírez-Mora et al. [77] carried out an SMS to investigate the current state of the research regarding productivity in the context of agile software development. According to the authors, a company's productivity in software development increases as more software can be developed with less time [77]. The authors based their research methodology on the recommendations of Kitchenham et al. [78]. After the authors performed the search process, they filtered the papers for relevance using pre-defined inclusion and exclusion criteria. No reference search was performed. A further step

in the research process was to carry out a quality assessment in order to only allow papers for the subsequent analysis that have a certain quality level [77].

Saldanha et al. (2019)

Saldanha et al. [79] conducted an SMS to determine the current state of research regarding the consideration of information security requirements in agile software development. According to the authors, information security is becoming more and more critical but may be neglected by the product owner when designing the product backlog. The authors based their research methodology on the recommendations of Peterson et al. [22, 80] from the year 2008 and the updated version from the year 2015 on. Later in the paper, the authors also mention that the definition of the search string needed to perform the search in the pre-defined databases was defined using the recommendations of Kitchenham et al. [71]. The authors classified all papers, which were published before the year 1999, as irrelevant. Furthermore, each paper had to have at least three pages and had to address security requirements and agile development in order to be relevant.

4. Methodology

This section describes in detail how the identification of high-quality publications relevant to large-scale agile software development research has been implemented on the one hand and on the other, how the consecutive mapping of the identified papers has been conducted. A detailed description of the process should enable other scientists to understand the results better and replicate the study. A visual overview of the process is provided with the help of Figure 1.2.

4.1. Identification and filtering of sources

This phase includes the identification of high-quality publications relevant to large-scale agile development research. For a systematic implementation of this phase, a search strategy was defined at the beginning. Subsequently, facets were derived from the research questions, which, together with the search strategy, influenced the definition of the search terms of the PS and MS. The PS was carried out to optimize the search terms of the MS and to find publications with the help of generic search terms which might not be found by more narrow search terms used in the MS. In addition to the search for publications in search engines (automated search), the PS ensured that studies which were known to the researchers in advance are also included.

4.1.1. Definition of research questions

A detailed description of the RQs is provided in Section 1.2.

4.1.2. Definition of search strategy

As described in the introduction of this chapter, the search for relevant literature follows a search strategy. This strategy is inspired by Zhang et al. [27] since they propose practical guidelines for the implementation of a search strategy, and therefore compensate what the guidelines for SLRs lack. Zhang et al. [27] conclude that implementing their suggestions will result in both an appropriate search strategy design and search strategy execution [27] . In order to design and execute an effective search strategy, Zhang et al. [27] advise researchers to answer the questions below:

1. Should the search approach be a manual search or an automated search?
2. Within which venues or databases should the search be executed?
3. Which part of the article should be searched? For what subject or evidence type should be searched?
4. What should be the structure of the search strings which are entered in the search engines?
5. When should the search be carried out? In what period should the search be carried out?

In the following, the questions defined above are answered in order. As a result of answering the questions, the search strategy of this thesis will be designed.

Should the search approach be a manual search or an automated search?
This SMS combines an automatic search with a manual search. While an automatic search allows searching the whole field efficiently with the help of search engines, the manual search within venues (e.g., journals or conferences) guarantees that no studies from highly relevant sources are lost [27]. Zhang et al. [27] argue that the quality of an automated search depends on the quality of the search strings, the capabilities of the search engines, as well as the diversity of the subject. Since the diversity of the subject cannot be influenced, only the quality of the search strings and sufficient capabilities of the search engines were ensured. Following a systematic approach to design the search terms ensured their effectiveness (see Section 4.1.4). Also, sufficient capabilities of the search engines were obtained by choosing only the most established search engines in the research community.

Which venues or databases should be searched?
As mentioned above, to obtain sufficient search capabilities only search engines that are heavily used in the research community were used. Further, the use of prominent search engines helped to increase the quality of the search results since they index venues of high relevance to large-scale agile software development. Table 4.1 presents the six search engines (DB1-DB6) that were chosen to be part of the automated search. During the preliminary search, it was assessed whether spending effort searching a specific database was worthwhile. Since the preliminary search results within all search engines were promising, all search engines were chosen for the main search.

ID	Search engine	Link	Selected
DB1	IEEE Xplore Digital Library	https://ieeexplore.ieee.org/Xplore/home.jsp	Yes
DB2	ACM Digital Library	https://dl.acm.org	Yes
DB3	Science Direct	https://www.sciencedirect.com	Yes
DB4	Web of Science	https://www.webofknowledge.com	Yes
DB5	Association for Information Systems (AIS) eLibrary	https://aisel.aisnet.org	Yes
DB6	Scopus	https://www.scopus.com	Yes

Table 4.1.: Search engines of the automated searches

Which part of the article should be searched?
Zhang et al. [27] suggest the article type (e.g., studies published in journals) and the subject as parts within a study that should be targeted by a search string. Whereas a search considering the article type can ensure that the set of resulting publications only contains high-quality publications, filtering for the topic ensures the relevance of the publications resulting from the search. In this SMS, article type, subject, and additionally, the abstracts of the publications were searched. The decision to also examine the abstracts was made during PS because, in this way the resulting studies were more relevant.

What should be the structure of the search strings which are entered in the search engines?
The search string was split into substrings, each covering a characteristic (e.g., agile software development), which marks a study as relevant for this SMS. These substrings were combined with logical operators. Since logical operators and strings can be applied within all search engines listed within Table 4.1, the chosen structure was suited for the execution of the search strategy. Since the syntax of the search engines differs during the PS, the search string was adopted to the syntax requirements of the search engines. Also, during the PS the substrings were optimized by a trial and error approach.

When and in what period should the search be carried out?
The search was carried out on Thursday, 3rd October 2019, since both the progress of the SMS and the author's knowledge required for the execution of the search strategy was sufficient at this point. The upper boundary of the period in which all publications fall was the date at which the search was carried out, and the lower boundary was 13th February 2001, which is the creation date of the Agile Manifesto.

4.1.3. Derivation of inclusion and exclusion criteria

Inclusion and exclusion criteria were defined to guide the decisions concerning the relevance of a publication during the filtering procedure. Also, these criteria influenced the design of the search strings, which were used within the search engines to conduct the automated search. The inclusion and exclusion criteria are represented by nine facets (F1-F9). Table 4.3 gives an overview of F1-F9 and the corresponding inclusion and exclusion criteria. Incorporating the feedback of leading scientists in the field of large-scale agile development ensured the effectiveness of facets to identify relevant studies. For a publication to be relevant the inclusion criteria of all facets must be true. The design of F1-F3 was inspired by the SLR of Dikert et al. [15], who also followed the guidelines provided by Kitchenham et al. [71]. Please note that only F1 and F2 were used to design the search string for the PS since some search engines provided a search string syntax which could not search for the publication type. Below, each facet is explained.

Facet 1: Agile software development
Likewise, Dikert et al. [15] examples of topics excluded by this facet are agile manufacturing as well as the implementation of agile methodologies in other environments than software engineering such as management boards. As a consequence, a publication is only relevant if it considers the application of agile methodologies in the software development context.

Facet 2: Large-scale
In this SMS, all projects were classified as large-scale when at least three agile teams were working on a joint solution. Therefore, this definition for large-scale differs from the taxonomy of Dingsøyr et al. [56], in which a project is considered a large-scale project if two to nine agile teams organize themselves in a forum in order to develop a joint solution. An example of such a forum would be Scrum of Scrums. This decision is based on the fact that the preliminary search has shown that many studies are relevant, but the agile teams are not organized in a forum. All publications which are based on projects that adhere to the definition of this SMS were included.

Facet 3: Affiliation to a pre-defined set of publication types
As mentioned above, some search engines did not have the capabilities to search for publications of certain types. Therefore, only the first two facets influenced the design of the search strings. After the search strings were entered into the search engines as part of the automated search, the options for filtering by publication type were revealed on the results page. In the next step, the publication type filtering options

of all the search engines were analyzed with a meta-analysis to get an overview of all filtering options (see Table 4.2). During the meta-analysis, the publication type categories were merged if they referred to the same publication type but had different names in the search engines. Afterward, the resulting categories were examined for their relevance. Then, relevant categories were included, and irrelevant categories were excluded. Ultimately all publications of the type conference, journal, and early access articles were considered relevant for the SMS. Other publication types lacked either scientific rigor or relevance to large-scale agile software development. Below, explanations of the categories' reviews, short communications, and early access articles are provided. Reviews referred to secondary research and were presented as a separate category by Science Direct, Web of Science, and Scopus. The category reviews was merged with the categories journals and conferences since every review was either be published in one of those categories. Short communications are a category that is used by Science Direct and contains only short papers. This category includes papers with less than three pages and is therefore excluded as described in facet nine. Early access articles, a category used by IEEE Xplore, are articles that are not available in the final electronic or print version. In some cases, they are not fully edited, but all of them are peer-reviewed and therefore included.

Filtering option	DB1	DB2	DB3	DB4	DB5	DB6	Relevance
Conference publication	✓	✓	✓	✓	✓	✓	Inclusion
Journal publication	✓	✓	✓	✓	✓	✓	Inclusion
Review	✗	✗	✓	✓	✓	✓	Inclusion
Early access article	✓	✗	✗	✗	✗	✗	Inclusion
Short communications	✗	✗	✓	✗	✗	✗	Exclusion
Conference review	✗	✗	✗	✗	✗	✓	Exclusion
Book chapter	✗	✗	✓	✓	✗	✓	Exclusion
Book review	✗	✗	✗	✓	✗	✗	Exclusion
Magazine	✓	✓	✗	✗	✗	✗	Exclusion
Newsletter	✗	✓	✗	✗	✗	✗	Exclusion

Table 4.2.: A meta-analysis of publication types

Facet 4: Language
Only publications written in English language were included. The reason for this is that English is the only language spoken by everyone involved in this research project. Also, no resources were available for translations.

Facet 5: Publication date
The upper limit for the included papers was the 3rd October 2019, and the lower boundary was 13th February 2001. A more detailed description of the reasons those dates were chosen is presented in Section 4.1.2.

Facet 6: Duplicates
Duplicates identified during the two searches were excluded. Furthermore, duplicates could occur within the PS and MS, since the six different search engines partially delivered the same publications. There were also duplicates between the PS and the MS. By excluding the duplicates the time required for the filtering process was considerably reduced.

Facet 7: Peer-reviewed publications
Only publications that underwent a peer-review were included in order to enhance the quality of the filter results.

Facet 8: Full-text access
Only publications that could be accessed with the rights provided by the Technical University Munich were included.

Facet 9: Conducted research papers
Only conducted research papers with at least four pages were classified as relevant to ensure that the resulting papers had enough substance. Opinion papers and experience reports were not classified as conducted research and therefore excluded.

ID	Facet	Inclusion criteria	Exclusion criteria
F1	Agile software development	The organization develops software with agile methods	The organization uses agile methods in other contexts (e.g., Agile manufacturing; Scrum in management boards)
F2	Large-scale	At least three agile teams are working on a joint solution	Less than three agile teams are working together, or the teams do not work on a joint solution
F3	Publication type	Conference publication, journal publications, reviews, early access articles	Newsletter, magazine articles, book reviews, book chapters, conference reviews, short communications
F4	Language	Publications in English language	Publications written in other words than the English language
F5	Publication date	Studies published in the period between 13th February 2001 and 3rd October 2019	Studies not published in the period between 13th February 2001 and 3rd October 2019
F6	Duplicates	Publications that are unknown to the filtering process	Publications that already have been processed
F7	Peer-reviewed publications	Publications that underwent a peer-review	Publications that did not undergo a peer-review
F8	Full-text access	Only publications that could be accessed with the rights provided by the Technical University Munich	Publications that could not be accessed with the rights provided by the Technical University Munich
F9	Conducted research papers	Publications with conducted research and with at least four pages	Publications with no conducted research (e.g. experience reports and opinion papers) and less than four pages

Table 4.3.: Inclusion and exclusion criteria of the filtering process

4.1.4. Definition of search terms

The search string of the PS was intentionally kept very generic. The generic design increased the probability of covering all relevant publications in the area of large-scale agile software development and enabled a quick establishment of an understanding of the properties of the search engines. The resulting generic search string is presented below:

Search term
large *scale* ***AND*** *agile*

The design of the MS search term was influenced on the one hand by an iterative improvement of the PS search term and on the other hand, inspired by Yang et al. [75], who carried out an SMS about the combination of software architecture and agile development. Just like Yang et al. [75], the search strings were first defined based on the topic of large-scale agile software development and RQs. Also, the PICO (Population, Intervention, Comparison, Outcomes) approach was used, which allows a systematic definition of search strings [71, 81]. Only the first component of the PICO approach, population, was applied. This decision was made because this SMS does not aim to restrict the population by interventions, comparison, or outcomes. Subsequently, the population is defined, and its search terms are derived. The population consists of three sets including search terms (see Table 4.4). The first set targets studies concerned with large-scale development projects, the second set targets studies considering agile software development, and the third set targets studies investigating scaling agile frameworks. The first two sets are connected with an "AND" operator. The third set is connected to the first two sets with an "OR" operator. In consequence, a study identified by the search term either considers large-scale and agile software development or a scaling agile framework and, therefore, is likely to be relevant. Based on the experiences of the researcher involved in this research project, it was decided to assign only the search terms *large-scale* and *scaling* to the first set. Like Yang et al. [75], the search terms of the second set include the agile methods identified in a literature review by Dybå et al. [74] and surveys on agile software development [75, 82, 83, 84]. Furthermore, the Dynamic Systems Development Method (DSDM) was neglected within the second set as it was proposed before the publication of the Agile Manifesto [85] (see Section 4.1.3). Also, the second set was limited to studies that consider agile methods in the software development domain. Therefore, likewise, Dikert et al. [15] this SMS excludes all publications dealing with manufacturing and not agile software development. With the help of the "AND-NOT" operator, all manufacturing strings were excluded from

the search results. The third set includes the scaling frameworks identified by Uludağ et al. [21] (see Table 4.5). The combination of the sets resulted in the following generic search string for the MS:

Search term

(Large-scale development ***AND*** *Agile software development)* ***OR*** *Scaling agile frameworks*

Set	Search term
Large-scale development	large-scale **OR** scaling
Agile software development	(agile **OR** agility **OR** extreme programming **OR** XP **OR** feature driven development **OR** FDD **OR** scrum **OR** crystal **OR** pair programming **OR** test-driven development **OR** TDD **OR** leanness **OR** lean software development **OR** lean development **OR** LSD) **AND NOT** manufacturing
Scaling agile frameworks	Crystal Family **OR** Dynamic Systems Development Method Agile Project Framework for Scrum **OR** Scrum-of-Scrums **OR** Enterprise Scrum **OR** Agile Software Solution Framework **OR** Large Scale Scrum **OR** Scaled Agile Framework **OR** Disciplined Agile 2.0 **OR** Spotify Model **OR** Mega Framework **OR** Enterprise Agile **OR** Delivery and Agile Governance Practice **OR** Recipes for Agile Governance in the Enterprise **OR** Continuous Agile Framework **OR** Scrum at Scale **OR** Enterprise **OR** Transition Framework **OR** ScALeD Agile Lean Development **OR** eXponential Simple Continuous Autonomous Learning Ecosystem **OR** Lean Enterprise Agile **OR** Framework **OR** Nexus **OR** FAST Agile

Table 4.4.: Overview of search sets and the corresponding terms

In Section 4.1.2, six search engines were chosen in which both PS and MS were performed. The search syntax of the six search engines differs. For this reason, the generic search terms of the PS and the MS were adapted to the requirements of the different search engines. The resulting search strings of the MS are shown in Table A.3. ScienceDirect does not offer the possibility to enter a search string, but a graphical interface with fields in which parts of the search strings can be entered. Because each field was limited to eight boolean operators the third set was limited to the three frameworks Large Scale Scrum, Scaled Agile Framework, and Disciplined Agile 2.0. The search interface of ScienceDirect is presented by Figure 4.1.

Scaling agile framework	Author	Organization	Publication year
Crystal Family	Alistair Cockburn	N/A	1992
Dynamic Systems Development Method Agile Project Framework for Scrum	Arie van Bennekum	DSDM Consortium	1994
Scrum-of-Scrums	Jeff Sutherland and Ken Schwaber	Scrum Inc.	2001
Enterprise Scrum	Mike Beedle	Enterprise Scrum Inc.	2002
Agile Software Solution Framework	Asif Qumer and Brian Henderson-Sellers	University of Technology	2007
Large Scale Scrum	Craig Larman and Bas Vodde	LeSS Company B.V.	2008
Scaled Agile Framework	Dean Leffingwell	Scaled Agile Inc.	2011
Disciplined Agile 2.0	Scrott Ambler	Disciplined Agile Consortium	2012
Spotify Model	Henrik Kniberg, Anders Ivarsson, and Joakim Sundén	Spotify	2012
Mega Framework	Rafael Maranzato, Marden Neubert, and Paula Heculano	Universo Online S.A	2012
Enterprise Agile Delivery and Agile Governance Practice	Erik Marks	AgilePath	2012
Recipes for Agile Governance in the Enterprise	Kevin Thompson	Cprime	2013
Continuous Agile Framework	Andy Singleton	Maxos LLC	2014
Scrum at Scale	Jeff Sutherland and Alex Brow	Scrum Inc.	2014
Enterprise Transition Framework	N/A	agile42	2014
ScALeD Agile Lean Development	Peter Beck, Markus Gärtner, Christoph Mathis, Stefan Roock and Andreas Schliep	N/A	2014
eXponential Simple Continuous Autonomous Learning Ecosystem	Peter Merel	Xscale Alliance	2014
Lean Enterprise Agile Framework	N/A	LeanPitch Technologies	2015
Nexus	Ken Schwaber	Scrum.org	2015
FAST Agile	Ron Quartel	Cron Technologies	2015

Table 4.5.: Overview of scaling agile frameworks by Uludağ et al. [21]

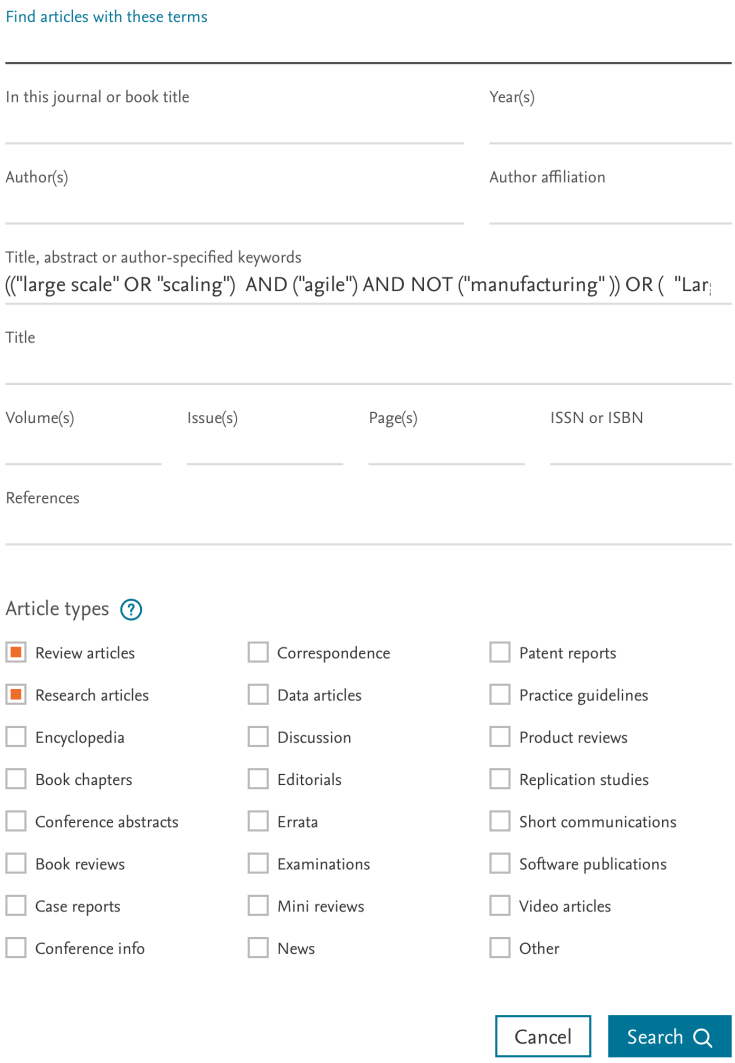


Figure 4.1.: The search interface of ScienceDirect

4.1.5. Overview of preliminary and main search

The following section describes the filtering process for relevant literature in large-scale agile development from the entry of search strings to the final list of relevant publications. Figure 4.2 gives an overview of the filtering process steps, the relationship between PS and MS, and shows how many studies are left after each performed step within the filtering process. Please note that the PS was performed before the MS. The

output of the filtering process consists of 108 studies, which were classified as relevant.

4.1.6. Execution of preliminary search

Within the PS, a manual and an automated search were implemented. The execution of a manual search had several objectives. The first goal was to ensure that relevant publications already known to the scientists in advance would be included. The second goal was to increase the quality of relevant studies resulting from the preliminary search, as this set was used to assess the effectiveness of the search string belonging to the automated search. Within the manual search, 72 studies were evaluated. These studies came from the SLR of Dikert et al. [15] and from the previous research of scientists involved in this project. For the automated search, the six search engines (see Table 4.1) were used as well as the preliminary search term, which was defined in Section 4.1.4. During the automated search, 146 studies were found. In conclusion, a total of 218 studies were included in the filter process of the manual search and the automated search. After removing the duplicates within the PS, 183 studies were passed to the next step, which involved the merging of the results of PS and MS.

4.1.7. Execution of main search

In contrast to the PS, the MS only consisted of an automated search. Here again the six search engines (see Table 4.1) were used for the automated search. However, the search terms for the MS defined in Section 4.1.4 were used. The result of the search engines included 2090 studies. After removing the duplicates within the MS, 1642 studies remained. These studies were merged with the result from the PS to remove the duplicates between the PS and the MS.

4.1.8. Duplicates between preliminary and main search

A total of 113 duplicates were identified between the PS and the MS. The ratio between the number of duplicates in this step and the number of studies representing the result of PS provides information on the effectiveness of the search term of MS. This ratio is presented below:

$$\frac{113}{183} \cdot 100\% \approx 61,75\%.$$

4.1.9. Filtering process of preliminary and main search

Figure 4.3 illustrates the filtering process performed by the scientists. Within the process the filter levels duplicate, full-text access, meta-data, abstract, full-text, final review,

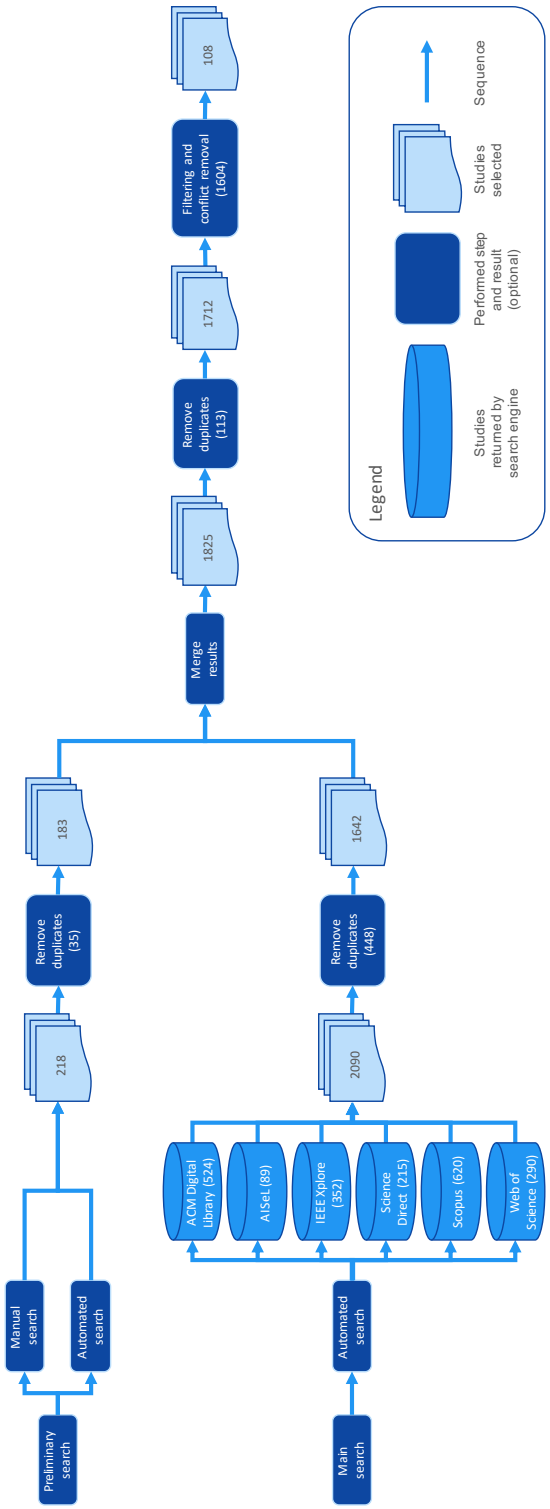


Figure 4.2.: Overview of the filtering process

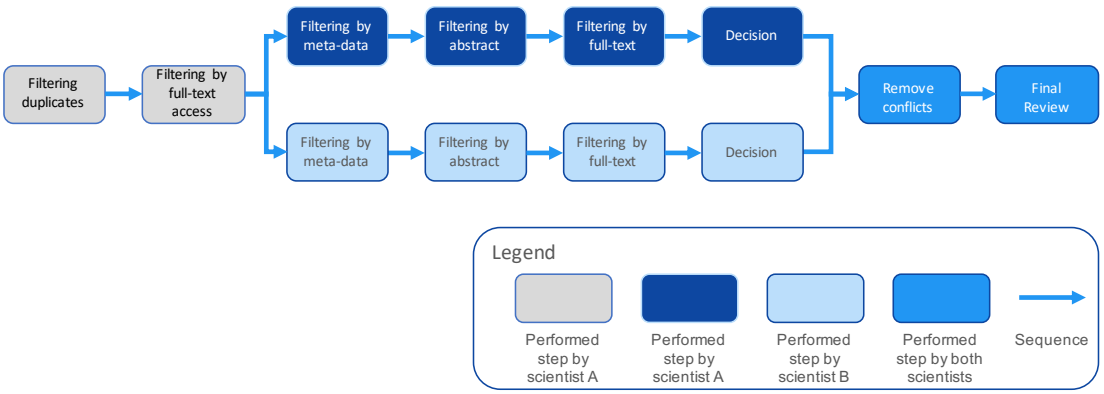


Figure 4.3.: Overview of the filtering steps

and final decision were introduced to increase the efficiency since for each filter level the effort to check the relevance (e.g., abstract analysis) was lower than the effort for an assessment in the following level (e.g., full-text analysis). A description of the individual levels is given in Table 4.6.

ID	Filtering level	Description
L1	Duplicate	Duplicates could occur between search engines of automated searches, between automated and manual search within the PS as well as between PS and MS. Duplicates were identified with the help of an Exel function to find duplicates based on a string comparison and additionally by a manual check conducted by both scientists.
L2	Full-text access	In the second step of the filtering process, it was checked whether the study could be accessed with the licenses of Technical University Munich. In sum, ninety-six studies could not be accessed.
L3	Meta-data (title, keywords)	In the third step, it was checked whether the title and the keywords of the study could be classified as relevant.
L4	Abstract	In the fourth step, it was checked whether the abstract of the study could be classified as relevant.
L5	Full-text	In the fifth step, it was checked whether the full text could be classified as relevant.
L6	Final Review	The final review has the same content as the result of the full-text analysis but was created to resolve the conflicts in a separate column.
L7	Final decision	The final decision is the final decision of whether a study was included or not.

Table 4.6.: Overview of filtering levels

Whereas L1 and L2 were performed by Scientist A alone, L3 to L6 were performed independently by the two scientists. This decision was made to prevent the two scientists from influencing each other's decisions. Level L7 was introduced to indicate whether a study was finally considered relevant after resolving possible conflicts. The process was implemented within an Excel spreadsheet (see Table 4.7). The four columns assigned to the scientist who was currently not performing the filter procedure were

hidden within the spreadsheet to prevent an influence of the active scientist by the assessments of the passive scientist. Table 4.7 contains the four possible cases which could occur during the process of the final decision (L6). Each case is described below. The first case is that both scientists agree that the study is relevant. All previous levels are also marked with the label inclusion. The second case is that both scientists agree that the study is irrelevant. The two scientists may have made this decision at different filtering levels. The third case is that both scientists disagree, and after a discussion, they decide to include the study. In the latter case, the scientists again disagree. After a discussion, they jointly make the decision to exclude the study. For all 102 conflicts, the two scientists were able to reach an agreement, so that no consultation of a third scientist was necessary. In total, the filtering procedure was performed for 1712 studies and 108 studies were classified as relevant.

Case	Description	L1	L2	L3 by Scientist A	L3 by Scientist B	L4 by Scientist A	L4 by Scientist B	L5 by Scientist A	L5 by Scientist B	L6 by Scientist A	L6 by Scientist B	L7
C1	Same decisions for L6 and study was included	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion
C2	Same decisions for L6 and study was excluded	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Exclusion	Exclusion	Exclusion	Exclusion	Exclusion
C3	Different decisions for L6 and study was included	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Exclusion	Inclusion	Exclusion	Inclusion
C4	Different decisions for L6 and study was excluded	Inclusion	Inclusion	Inclusion	Inclusion	Inclusion	Exclusion	Inclusion	Exclusion	Inclusion	Exclusion	Exclusion

Table 4.7.: The four conflict cases within L6 and Excel schema

4.1.10. Selected studies

The selected studies are listed in Appendix A.1.

4.2. Data extraction and systematic mapping

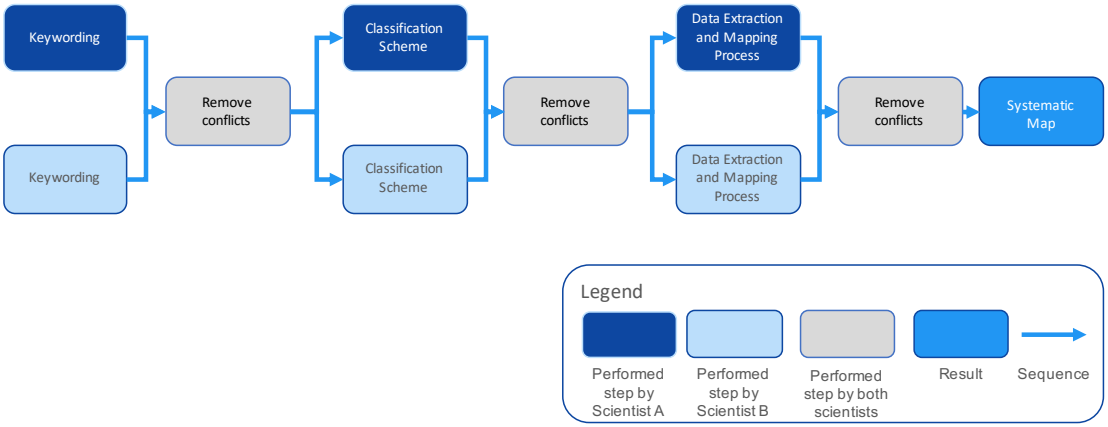


Figure 4.4.: Data extraction steps adopted from Petersen et al. [22]

Figure 4.4 illustrates the steps of the data extraction process, which was performed multiple times to create a systematic map for thirteen categories. The thirteen categories (C1-C13) are described in Table 4.8. The input for each iteration of the process were the 108 studies (see Table A.1), which were identified as relevant during the filtering process. Similarly to the filter process, both scientists performed the data extraction process separately to prevent mutual influence. The first step in the process, the keywording, was carried out if the categories could not be classified directly, or no classification scheme could be found in the literature. Thus, a keywording was performed for the categories C5, C6, C7, and C13. During the keywording, each scientist selected the words, which describe the properties of a study for the respective category best. As mentioned above, both scientists performed this step separately. After both scientists finished the keywording step, conflicts between the scientist were resolved. Then, both scientists worked separately with the final keywords to derive a classification scheme. Again, deviations between the two classification schemes were resolved. In the last step, each scientist assigned the studies to the final classification scheme. After all assignment conflicts were resolved, the mapping was final. Similar to the filtering process, an Excel spreadsheet was used for the data extraction process in which the columns of the inactive scientist were collapsed to avoid psychological bias. In the

following, the classification schemes of the categories are described.

ID	Category	Description	Keywording
C1	Author	This category contains all the authors of a study. The authors were identified using the BibTeX file of the study. No keywording was done because the authors could be classified directly.	✗
C2	Cite count	The cite count was determined with the help of google scholar. No keywording was needed because the cite count could be classified directly.	✗
C3	Year	The year was identified with the help of Google Scholar. No keywording was needed because the year could be classified directly.	✗
C4	Country	The country was taken from the full-text of the study. Decisive was the country in which the institution of the first author was located. No keywording was needed because the country could be classified directly.	✗
C5	Exact publication channel	The publication channel name was taken from the BibTeX file. A keywording was carried out.	✓
C6	General publication channel	The publication channel type was taken from the BibTeX data. A keywording was performed.	✓
C7	Topic	The topic was taken from the keywords, the abstract, and, if required, from the full-text. A keywording was carried out.	✓
C8	Research type	The research type was taken from the keywords, the abstract, and, if required, from the full-text. No keywording was needed because an existing classification scheme was used.	✗
C9	Research approach	The research approach was taken from the keywords, the abstract, and, if required, from the full-text. No keywording was needed because an existing classification scheme was used.	✗
C10	Research contribution	The research contribution was taken from the keywords, the abstract, and, if needed, from the full-text. No keywording was needed because an existing classification scheme was used.	✗
C11	Primary and secondary research	The classification into primary and secondary study was taken from the keywords, the abstract, and if required from the full-text. No keywording was needed because the primary and secondary studies could be classified directly.	✗
C12	Rigor and relevance	The ratings for rigor and relevance were taken from the full-text. No keywording was needed because the ratings for rigor and relevance studies could be classified directly.	✗
C13	Research agenda	The research agenda was taken from the full-text. A keywording was performed.	✓

Table 4.8.: Overview of data extraction categories

4.2.1. C1-C4: Author, cite count, year and country

The authors, citations, years, and countries were extracted from the studies and classified without a keywording. The classification scheme for authors is a list containing all authors. Since no author name occurred several times, the original names could be used. The classification scheme of the years and the citation count is based on the positive natural numbers, including zero. The classification scheme of the countries contains all country names belonging to the institutions, for which the first authors of the 108 studies research.

4.2.2. C5: Exact publication channel

For keywording, the exact publication channel was usually taken from the BibTeX files, otherwise from the full-text. Because the keyword could be copied and did not have to be designed, there were no conflicts. If the keyword contained not only the name of the publication channel but other information, the additional information was removed from the keyword to create the classification scheme. For instance, the keyword for S4 is "Twenty-fourth Americas Conference on Information Systems, New Orleans, 2018" and the keyword for S42 is "Twenty-fifth Americas Conference on Information Systems, Cancun, 2019". After removing the unnecessary information in the keywords, both studies were assigned to the category Americas Conference on Information Systems in the classification scheme.

4.2.3. C6: General publication channel

For the general publication channel, the classification scheme consisted of the three categories journal, conference, and workshop. In general, the information about the underlying category of the publication channel was taken from the BibTeX file. If the information was not contained in the BibTeX file, it was retrieved from the full-text.

4.2.4. C7: Topic

After the keywording, the studies S1-S108, were arranged in a tree graph in which nodes represent the research streams of large-scale agile development (see Figure 5.24). Along a branch (several nodes connected by edges represent a branch), the research topics become more specific. In other words, the higher the depth of a node, the more specific the topic represented by that node. If in a study several topics were covered, the study was assigned to multiple nodes. The naming of the nodes was adjusted continuously so that the tree best represents the previously classified studies. The classification scheme of the topic consists of the final set of nodes.

4.2.5. C8: Research type

Wieringa et al. [86] inspired the classification scheme for the research type. The relevant studies of these SMS fell into the three categories Evaluation research, Philosophical papers, and Solution proposal. These categories are described in Table 4.9. Categories validation research, conceptual proposals, opinion papers, and experience papers proposed by Wieringa et al. [86] were excluded. The relevant studies could not be distributed among these categories and, therefore, were dismissed.

Research type	Description
Evaluation research	Techniques are used in companies, and an evaluation of the technique is carried out. Therefore, it is shown how the technique is implemented in practice (solution implementation) and which effects the implementation has with regard to benefits and drawbacks (implementation evaluation). This also involves the identification of problems in the industry.
Philosophical papers	These papers present a new perspective on existent things by organizing the domain into a taxonomy or into a conceptual framework.
Solution proposal	There is a proposed solution to a problem, the solution can be either novel or a significantly enhanced version of an existent technique. The benefit and applicability of the solution is demonstrated by a small example or argumentation.

Table 4.9.: Classification scheme of research types, as proposed by Wieringa et al. [86]

4.2.6. C9: Research approach

Table 4.10 presents the classification scheme for the research approach which is inspired by Berg et al. [87] and Rodríguez et al. [25] and has been extended by two further approaches. Berg et al. [87] describe the research approach Design and creation. Rodríguez et al. [25] describes the approaches Case study, Survey, Action research, Grounded theory, Mixed methods, and a category called Not applicable to which all studies were assigned that did not fit into other categories. The other categories Theoretical and SLR/SMS, were added to the classifications schema because, in the filtering process, their necessity was recognized.

4.2.7. C10: Research contribution

The classification scheme for the contribution type is inspired by Paternoster et al. [88] and is described in Table 4.11. In contrast to Paternoster et al. [88], this SMS neglects the two contribution types Tool and Implication, since the 108 studies could not be assigned to them.

4.2.8. C11: Primary and secondary studies

This classification scheme consists of two categories: primary studies and secondary studies. No keywording was performed.

4.2.9. C12: Rigor and relevance

The classification scheme for the rigor and relevance categories was adopted from Ivarsson and Gorschek [89]. The rigor of a study (see Table 4.13) is evaluated by the

Research approach	Description
Survey	If the study collects quantitative and/or qualitative data with the help of a questionnaire or interviews. The sample has to be representative to generalise to the entire population.
Design and creation	The development of a new IT product or artifact, or new model/method.
Case study	If one of the following conditions is fulfilled: 1) The study states one or more research questions and some of them or all are answered with a case study. 2) The study evaluates a theoretical concept empirically by implementing it in a case study.
Action research	When the study explicitly states this research method.
Grounded theory	When the study explicitly states this research method.
Mixed methods	If the study uses multiple data collection methods.
Not applicable	Either if study does not define the applied research method or and it cannot be inferred or interpreted from study.
SLRs or SMSs	When the study explicitly states this research method.
Theoretical	If the study is of a theoretical nature but not explicitly describes that a grounded theory approach was used.

Table 4.10.: Classification scheme of the research approach inspired by Berg et al. [87] and Rodríguez et al. [25]

Contribution type	Description
Model	Presentation of an observed reality with the help of concepts or related concepts resulting from a conceptualization process.
Theory	Establishing relationships between cause and effect from determined results.
Framework or methods	All Models that relate to the design of software or to the management of development processes.
Lessons learned	List of outcomes, which is analyzed directly from the research results obtained.
Guidelines	A list of advice represents a summary of the research results obtained.

Table 4.11.: Classification scheme of the contribution type inspired by Paternoster et al. [88]

Aspect	Contribute to relevance (1)	Do not contribute to relevance (0)
Subjects	The persons involved in the assessment are representative of the users who will use the technology in reality. An example of a person fulfilling this criterion is an industry expert.	The subjects used in the evaluation are not representative of the users who will use the technology. Examples for subjects that do not contribute to relevance are students and scientists. Also, if the subject is not mentioned, a 0 will be assigned.
Context	The setting of the evaluation is representative of the usage setting in practice. An example of setting fulfilling this criterion is an industry setting.	The setting in which the evaluation is performed is not comparable to the intended usage setting. An example of setting fulfilling this criterion is a laboratory environment.
Scale	The applications which are used in the evaluation are of a size that is comparable to the size of applications used in practice.	The applications which are used in the evaluation are, for example, toy or downscaled industrial applications, which are of a size that is nor comparable to the size of applications used in practice.
Research method	The research method which was conducted is one that fosters the investigation of realistic situations. Examples for such research methods are action research, lessons learned, case study, field study, interviews, or descriptive and exploratory surveys.	The research methods are not investigating real situations. Examples for such research methods are conceptual analysis or mathematical, and laboratory experiments.

Table 4.12.: Classification scheme of the relevance inspired by Ivarsson and Gorschek [89]

aspects context, study design, and validity. For each aspect, the scope of its description is assessed. Based on the scope, a decimal number is assigned to the aspect. The scope can be strong (1), medium (0.5), or weak (0). In conclusion, the sum of the three aspects can be a decimal number between 0 and 3 and determines the relevance of the study. Likewise, the assessment of the relevance of a study (see Table 4.12) is based on aspects. However, the assigned value is based on an ordinal scale. Therefore, each of the four aspects subjects, context, scale, and research method, can be declared relevant (1) or irrelevant (0). The sum can be a positive natural number between 0 and 4 and determines the relevance of a study.

4.2.10. C13: Research Agenda

For the classification scheme of the research agenda, the tree scheme of the topics was used. However, in this scheme, only the nodes (topics) with the depth one were

Aspect	Strong description	Medium description	Weak description
Context described	The context is described in detail in the study so that it can be understood by the reader and is comparable with other contexts. The description of the development mode is important for the software development domain.	The context is described, but the scope of the description is not large enough for the reader to understand or compare the context.	No description of the context exists.
Study design described	The study design must be described in detail so that the reader can understand it. To achieve this, for example, the measured variable or the controls used can be described.	The design of the study is only briefly described.	There is no description of the design of the study.
Validity discussed	The validity of the research carried out is discussed in detail. The threats to validity are listed, and the actions taken to prevent them are described.	The validity of the research is mentioned but not described in detail.	There is no consideration of validity.

Table 4.13.: Classification scheme of the rigor inspired by Ivarsson and Gorschek [89]

used because the full scheme was too detailed, and therefore to some topics no future research areas could be assigned. The studies were searched for the string "future" in order to efficiently identify the text passages providing information on worthwhile future research projects. Irrespective of the filtering process of the relevant studies, nine workshops were selected, which contain promising proposals for future research. The workshops are listed in Appendix A.2. The proposed research directions were then reformulated as questions and assigned to the corresponding topic (node) in the tree.

5. Results

This section aims to answer RQ1 to RQ12 with frequency analyses and other descriptive statistics. These statistics are synthesizing the data that emerged from the mapping of the selected studies into the classification schemes (C1-C13). Each statistic is based on a population of 108 studies that were identified during the filtering process. A detailed description of the filtering and mapping is presented in Chapter 4.

5.1. Distribution of publications per year

> ***Research question 1:*** *What is the distribution of publications per year in the area of large-scale agile development?*

To answer the first research question, the publication year of each study was determined. After retrieving the publication year of the studies, two statistics were calculated. The first statistic presents the count of the number of studies published in the same year. The second statistic displays the cumulative sum. Figure 5.1 shows both statistics. The 108 studies are spread over a period between 2008 and 2019. On average, nine studies were published each year. The number of publications fell in 2009 and 2010. In 2010, two studies were published, and thus the fewest studies during the 12 years. However, the number of publications rose from 2013 onwards. The increase in the number of publications is particularly noticeable between 2016 and 2019, when the total number of publications more than doubled from 53 to 108 studies. The peak number of published studies for one year is 23 studies and was reached in 2018. The strongest increase in the number of published studies took place in 2018 when eight more studies were published than in the previous year. The sharpest decline in the number of studies was in 2015 when seven studies less were published than in the previous year. In the last three years (2017-2019), Germany (16 studies) has been the country with the most publications. Norway (10 studies) ranks second, followed by Sweden (7 studies). The average number of publications in journals rose by 286.27% in the last three years to 6.67 studies/year (2017-2019) compared to 2.33 studies/year (2014-2016) in the previous three years. The increase in the number of conferences was 168.86% from 5.33 studies/year (2014-2016) to 9 studies/year (2017-2019). Every year most of the studies were of the type evaluation research. In some years, no studies of

the category Philosophical papers and Solution proposal were published. Between 2014 and 2019, the number of published philosophical paper studies decreased in four out of five years (2015, 2016, 2017, and 2019 decreased the number). In each of the twelve years, there were always more case studies than studies with other research approaches. Since 2013, the number of studies with the contribution type lessons learned has risen sharply. Accordingly, the average number of publications providing lessons learned was 9.29 studies/year (2013-2019) compared to the average of 2.6 studies/year of all previous years.

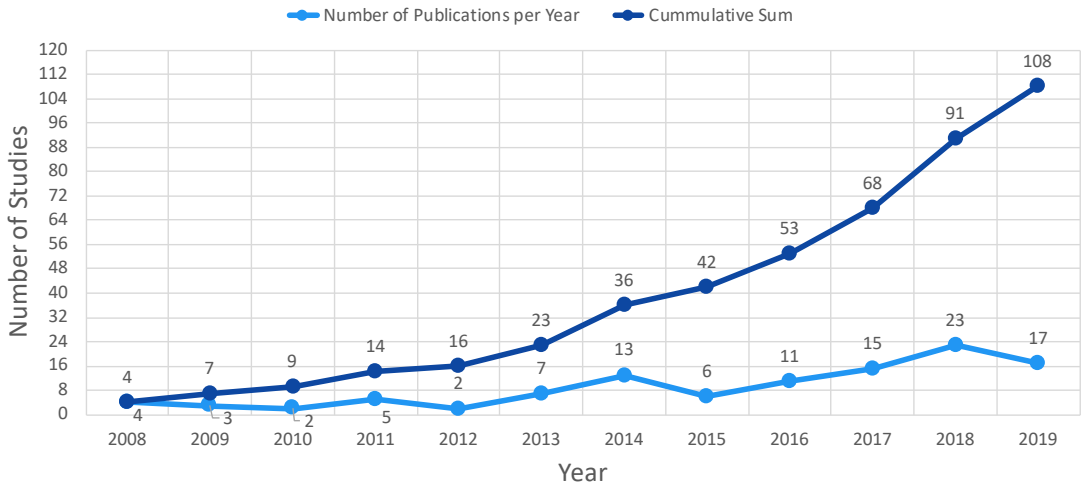


Figure 5.1.: Distribution of publications per year

5.2. Geographical distribution of studies

> ***Research question 2:*** *How are studies in the large-scale agile development research geographically distributed?*

To answer the second research question, the countries in which the research institutions of the first authors are located were identified. After the mapping between the studies to the respective countries, the count a single country appeared in the set of selected studies was determined. Subsequently, one statistic was calculated, which was used in two figures. The first figure presents the count of the number of studies published in the same country with the help of a bar chart (see Figure 5.2). The other one shows the same numbers with the help of a map chart (see Figure 5.3) and assigns the studies to regions and continents. In the following, both figures will be described in detail. In Figure 5.2, the 21 countries and the number of studies published by the authors

from these countries are presented. The country where the largest number of studies originate is Finland (21 studies), followed by Germany (20 studies), Sweden (18 studies), and Norway (16 studies). The total number of publications from these four countries is 75, representing 69.44% of all selected studies. In total, ten countries (47.62%) have published only one study. Finland (1332 citations) is the country that has received the most citations. The second most cited country is Sweden (680 citations) before Norway (300 citations). Germany has published six solution proposal studies and thus at least three times as many studies with this research type as any other country. While there are five studies from the United Kingdom with a grounded theory approach, all other countries have, at most, one study with this approach. Figure 5.3 shows that Northern Europe is the region where most studies originate. The sum of publications from Northern Europe (Finland, Germany, Sweden, Norway, the United Kingdom, Netherlands, Denmark, and Ireland) amounts to 91 and thus 84.3% of the selected studies. The eight Northern Europe countries account for 38.1% of all 21 countries where the selected studies were published. A total of 98 studies originate from Europe, seven from Asia, one each from the continents of North and South America, and Australia. Africa is the only continent with no publication.

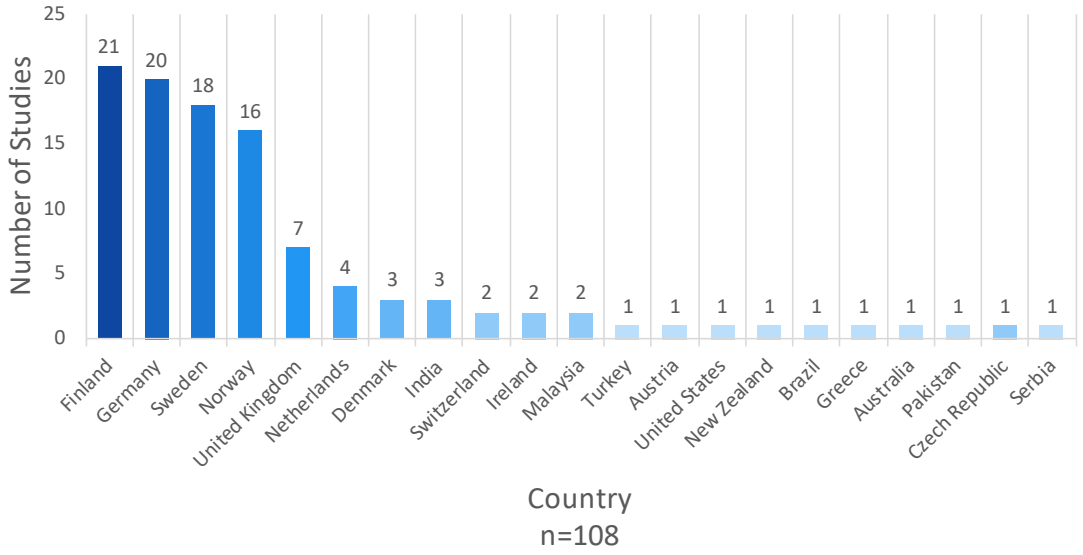


Figure 5.2.: Geographical distribution of studies

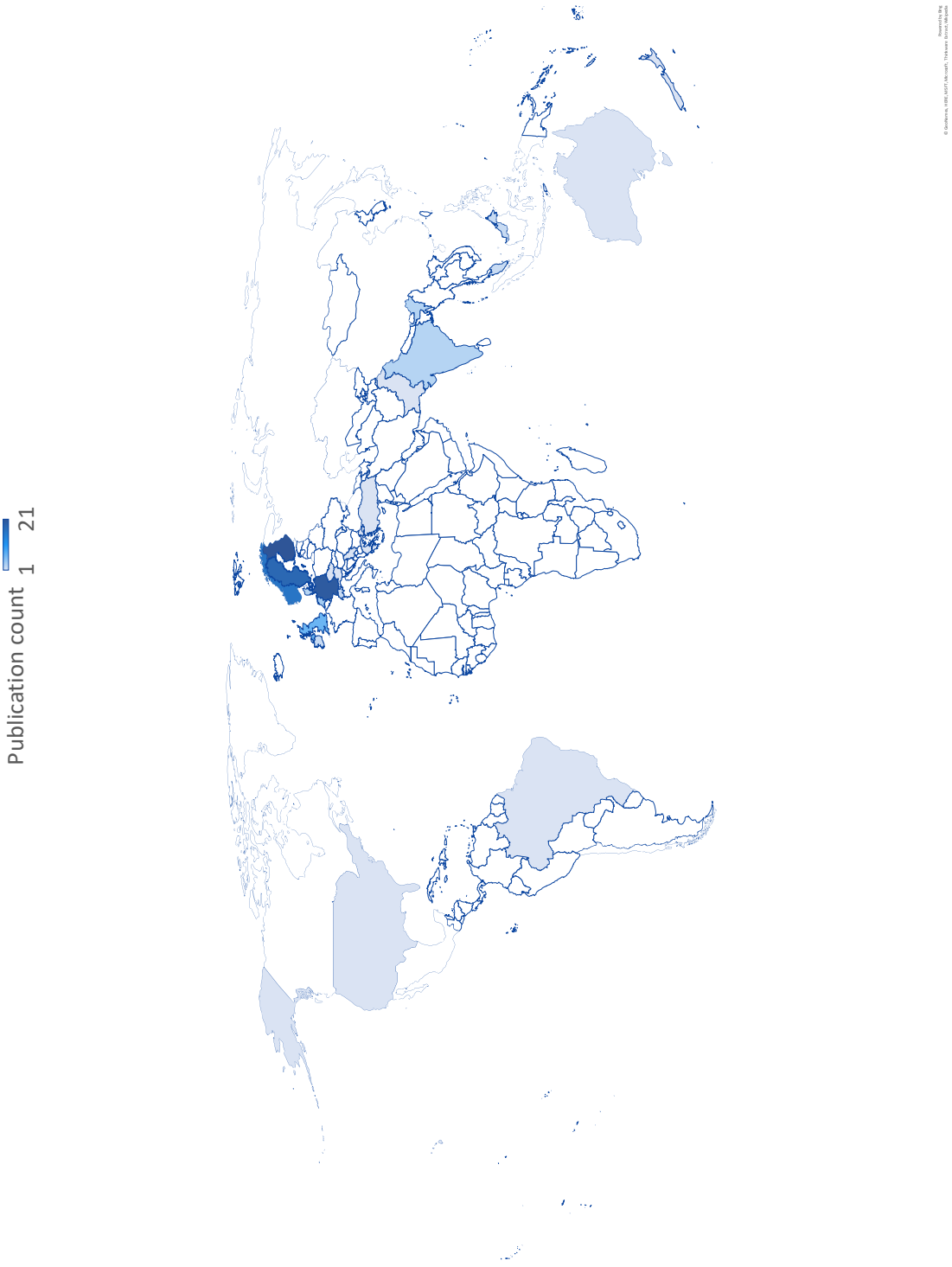


Figure 5.3.: Geographical distribution of studies by regions and continents

5.3. Most salient studies

Research question 3: *Which studies are the most influential in terms of number of citations?*

To answer the third research question, the number of citations was extracted from each study, and the average number of citations per year was calculated. Figure 5.4 shows the 25 studies with the most citations. The publication S23 (254 citations) has the highest total number of citations, followed by S6 (249 citations), S1 (222 citations), and S2 (219 citations). Study S23 (∅ 63.5 citations/year) also has the highest yearly average of citations ahead of the studies S44 (∅ 38.5 citations/year), S2 (∅ 24.33 citations/year), and S6 (∅ 20.75 citations/year). The studies S44, S19, S10, and S45 are not among the top ten studies in terms of the total number of citations but are among the top ten with the highest yearly average of citations. On average, each of the 108 selected studies has a citations count of 30.45 and 5.66 citations per year. The 25 studies with the most citations have a share of 71.36% of all citations. For 45 studies (41.67%), the average citation number is less than three citations per year. On average, studies with the publication year 2008 (139.25 citations/year) have the most citations, followed by studies from 2009 (94.33 citations/year) and studies published in 2010 (88.5 citations/year). Among the countries that can be assigned at least three studies, Finland (63.43 citations/year) has, on average, the most citations, followed by Sweden (37.78 citations/year) and the United Kingdom (26.86 citations/year). Of the studies published in journals, the study S23 (254 citations) received the most citations ahead of S6 (249 citations) and S1 (222 citations). Among conference studies, the study S35 (118 citations) had the most citations before S55 (88 citations) and S68 (64 citations). S5 (88 citations) is the study with the highest number of citations within the workshop publications followed by S56 (15 citations) and S97 (11 citations). The average number of citations calculated for the three research types evaluation research (30.78 citations/study), philosophical papers (27.08 citations/study), and solution proposal (31.46 citations/study) are similar. Among the top 25 studies, half of the studies are case studies. In the top 25 studies, 75% have the contribution type lessons learned. The only two secondary research studies in the top 25 are the studies S23 and S10.

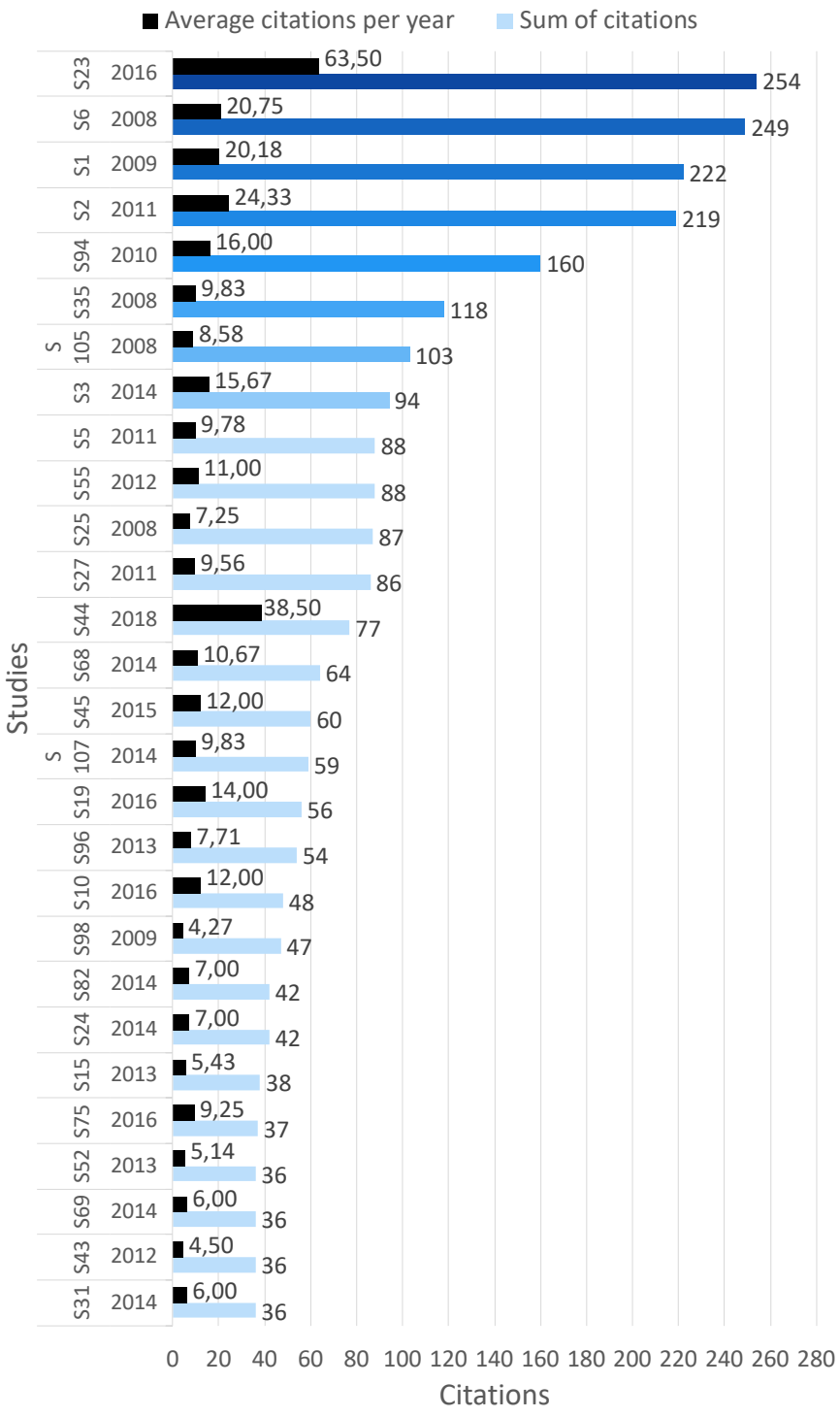


Figure 5.4.: Most salient studies

5.4. Most salient scientists

Research question 4: *Who are the most salient scientists in the field of large-scale agile development measured by number of publications and citations?*

To answer the fourth research question, two diagrams are described, which represent the 25 most salient authors in terms of their publication (Figure 5.5) and citation counts (Figure 5.6). For both diagrams, the top 25 authors were identified from a total of 172 authors. Figure 5.5 presents the 25 most salient authors measured by their publication counts. The average publication count of the top 25 researchers is 5.21 studies, the minimum is three studies, and the maximum is 16 studies. The maximum value was reached by Paasivaara (16 studies), followed by Lassenius (15 studies) and Uludağ (10 studies). With 16 publications, Paasivaara is the author of 14.82% of the 108 studies. Uludağ (∅ 3.33 publications/year) publishes the most publications on average followed by Matthes (∅ 3 publications/year) and Kleehaus (∅ 2 publications/year). Out of all scientists, the data shows that 144 (83.72%) scientists published less than three studies, and 164 (95.3%) scientists did publish less than six times. The average experience of the top 25 researchers is 6.29 years, whereas the average experience of authors with less than six publications is 4.68 years, and that of authors with less than three publications is 4.49 years. In the second diagram (Figure 5.6), the 25 most salient authors were selected based on their citation count. Among the 25 most frequently cited authors, the average number of citations is 246; the minimum 86 citations, and the maximum 920 citations. Here again, Paasivaara obtained the maximum number, followed by Lassenius (890 citations) and Wohlin (415 citations). Both Paasivaara and Lassenius hold a share of over 27% of the total number of citations calculated for the 108 selected studies. Wohlin, the third most cited author, holds a share of 12.62%, which is less than half of the citations of Paasivaara and Lassenius. The most cited author, on average, is Paasivaara (∅ 76.66 citations/year). The second-highest mean is held by Lassenius (∅ 74.17 citations/year), followed by Dikert (∅ 41.43 citations/year). With regard to all scientists, the data shows that 94 (54.65%) scientists have less than 20 citations, and 126 (73.26%) scientists were cited less than 40 times. While the mean experience of the top 25 researchers is 9.12 years, the average experience of authors with fewer than 40 citations has been 3.63 years, and of authors with fewer than 20 citations was 3.17 years. Please be advised that the researches Paasivaara and Lassenius, as well as Uludağ, Matthes, and Kleehaus are on most of their papers co-authors.

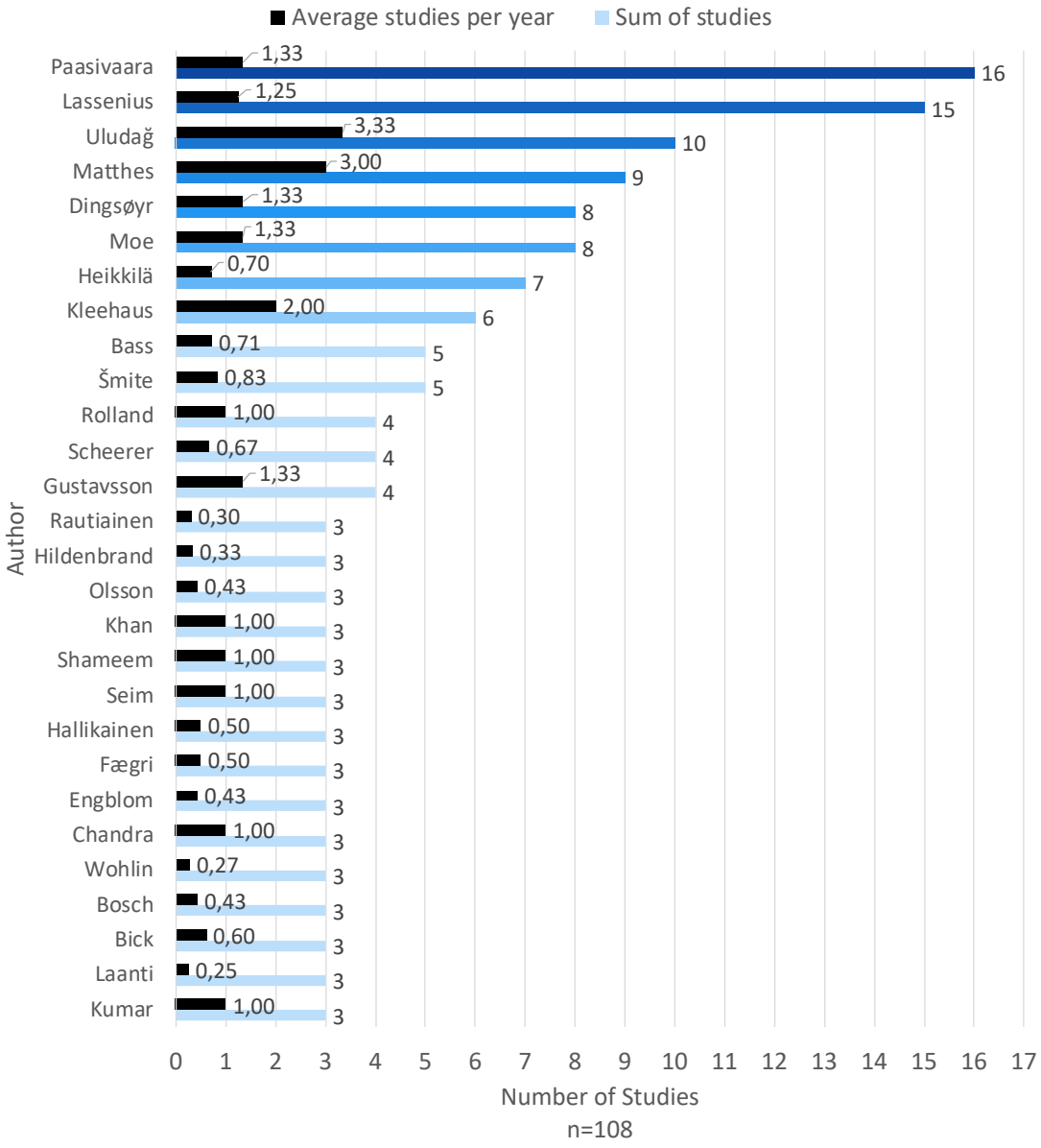


Figure 5.5.: Most salient scientists by the number of publications

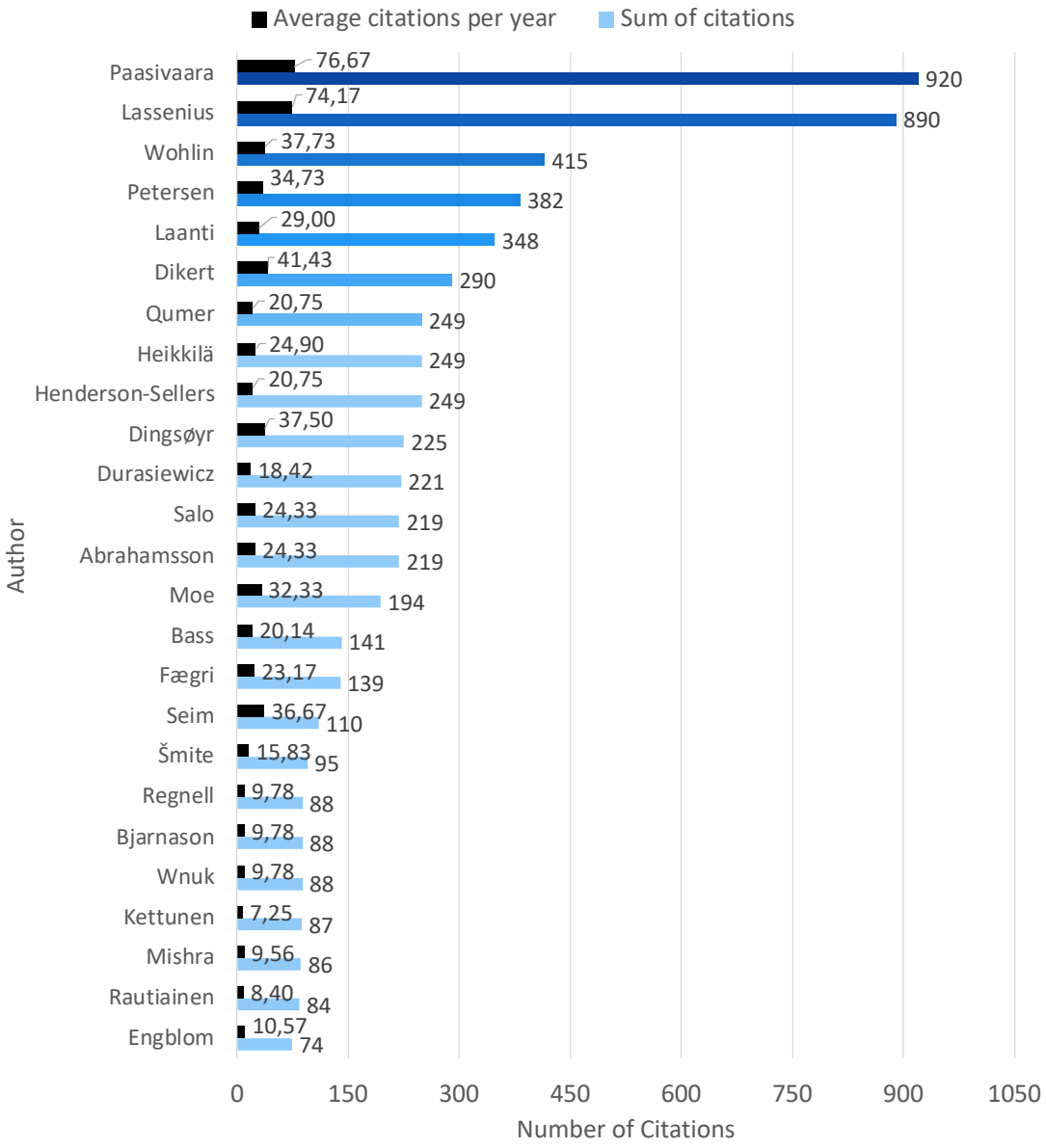


Figure 5.6.: Most salient scientists by the number of citations

5.5. Publication channels

Research question 5: *What are publication channels of studies related to large-scale agile development?*

Figure 5.7 shows the number of publications for the three publication channels. Each of the 108 selected studies was published in one of these three channels. Most studies were published in conferences (57 studies), followed by journal publications (35 studies) and publications in workshops (16 studies). The first studies in conferences and journals were published in 2008, whereas the year of the first workshop publication is 2011. Journal publications have on average the most citations (∅ 58.09 citations/study), followed by conferences (∅ 19.05 citations/study) and workshops (∅ 6.69 citations/study). Publications in journals have been increasing since 2015, and in workshops, they have been decreasing since 2016. In all publication channels, most studies are of type evaluation research. The most similar proportions between the research types exist in the conferences. Here Philosophical papers (15.79%) and Solution proposal (12.28%) have their largest proportions among all channels. Only 10% of studies with a grounded theory approach are published in workshops. More than half of the case studies (59.02%) are published in conferences. Of the studies which have a model as contribution type, 70% are published in conferences. The assignment of the 43 venues to conferences (Figure 5.8), journals (Figure 5.9) and workshops (Figure 5.10) is described below. Only the black bars are relevant for this section. In Section 5.10, the venues are assessed according to their rigor and relevance. The category conferences consist of 23 venues. The two venues with the highest number of publications are the International Conference on Global Software Engineering (11 studies) and the International Conference on Agile Software Development (10 studies). The proportion of venues with one publication on all 23 conference venues is 60.87% and, therefore, almost identical to the same proportion within the journals (60%). The journal studies were published in fifteen venues. The three venues with the most publications are IEEE Software (6 studies), Information and Software Technology (6 studies), Empirical Software Engineering (5 studies), and Journal of Systems and Software (4 studies). In total, five workshops were identified. The workshop with the most publications is the International Workshop on Large-Scale Agile Development (12 studies). All other workshops have only one publication. As a consequence, the proportion of venues with one workshop on all workshops is 80%.

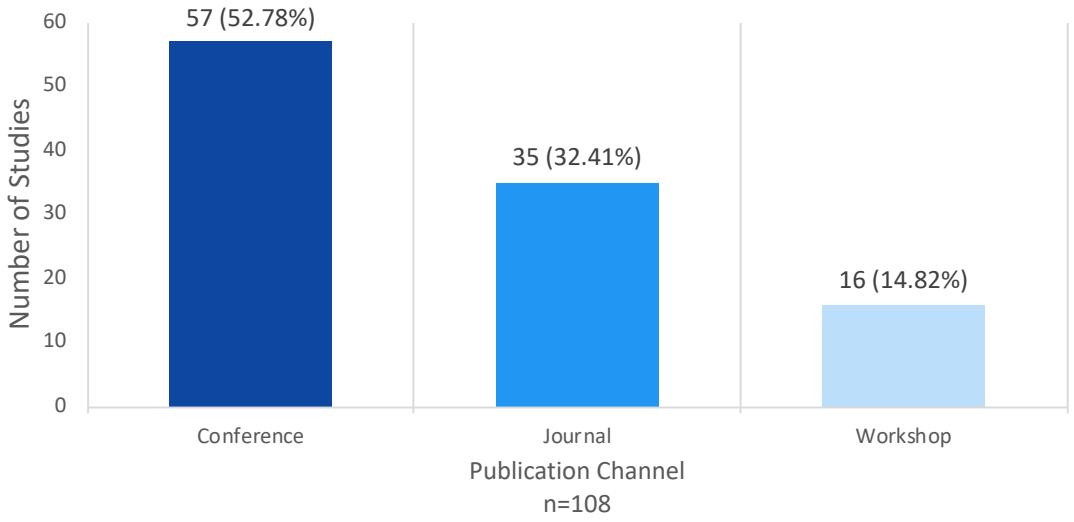


Figure 5.7.: Distribution of studies in publication channels

5.6. Research types used in the studies

Research question 6: *What research types of the studies have been used in large-scale agile development?*

Figure 5.11 shows the number of publications for the three research types of evaluation research (83 studies), solution papers (13 studies), and philosophical papers (12 studies). Each of the 108 studies was assigned to one of the three types. More than six times as many papers of the type evaluation research were published than papers of other research types. Between 2017 and 2019, the average number of evaluation research studies was 14.67 studies/year, thus above the average of all previous years (∅ 4.33 studies/year). The first studies of the type evaluation studies and solution proposals were published in 2008, whereas the first philosophical paper was published in 2011. Solution papers had, on average, the highest citation count (31.46 citations/study), followed by evaluation research (30.78 citations/study) and philosophical papers (27.08 citations/study). Altogether 77.68% of the citations can be assigned to the research type evaluation research. The remaining citations are almost equally distributed between the philosophical papers (9.88%) and the solution proposals (12.44%). The researchers from Finland (19 studies), Sweden (17 studies), Norway (13 studies), and Germany (10 studies) published 54.63% of the papers that can be assigned to type evaluation research. The scientists from Germany are leading both other research

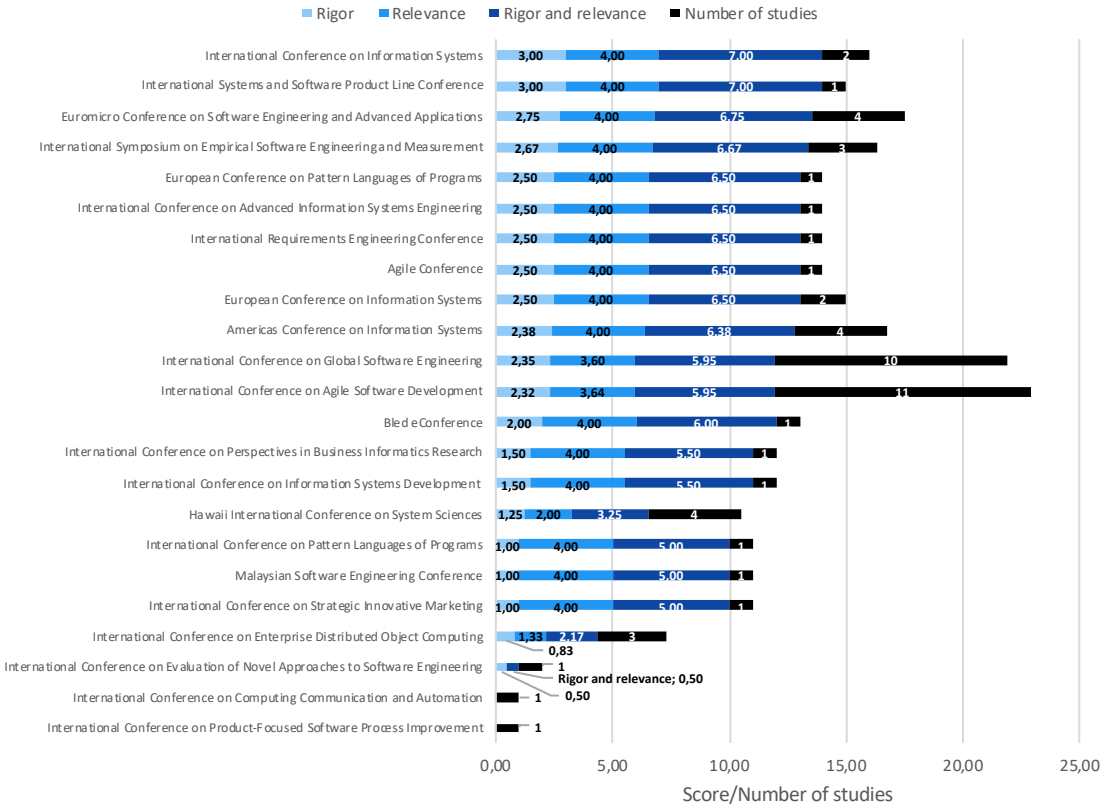


Figure 5.8.: Distribution of studies in conferences

types of philosophical papers (4 studies) and solution proposals (6 studies). All three research types can be found most frequently in the publication channel conferences. In the case of philosophical papers, 69.23% of all studies were published in conferences. Most studies (65.06%) of the research types evaluation research contain the research approach case study. For the research type solution proposals the most commonly used research approach is design and creation (i.e., 38.46% of the solution proposals). Evaluation research most frequently results in providing lessons learned (90.36%). The contribution type models is the most common in philosophical papers (66.67%) and framework/methods are the most usual result in solution papers (61.54%). Secondary studies account for 10.84% of all studies of type evaluation research. The studies of the other types do not fall into the category of secondary studies.

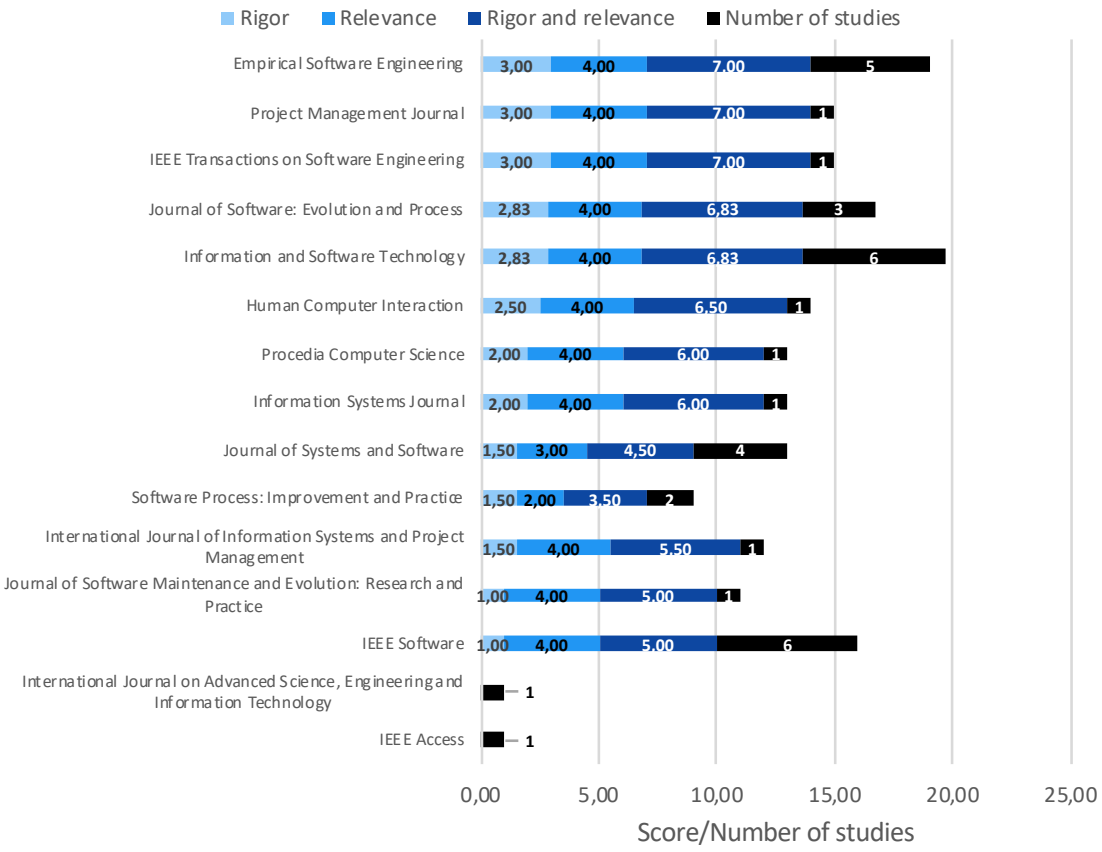


Figure 5.9.: Distribution of studies in journals

5.7. Research approaches used in the studies

Research question 7: *What research approaches have been used in studies related to large-scale agile development?*

Figure 5.12 shows the number of publications for the eight research approaches and the category not applicable. Each of the 108 studies was assigned to one of these approaches, or the category not applicable. Case studies (61 studies) are the most frequently implemented research approach, followed by grounded theory (10 studies), SLR/SMS (9 studies), and mixed methods (9 studies). The three approaches grounded theory, SLR/SMS, and mixed methods were almost equally frequent used. Case studies were used more than six times as often as all other research approaches and accounted for 56.48% of all approaches. During the last three years, an average of ten case studies

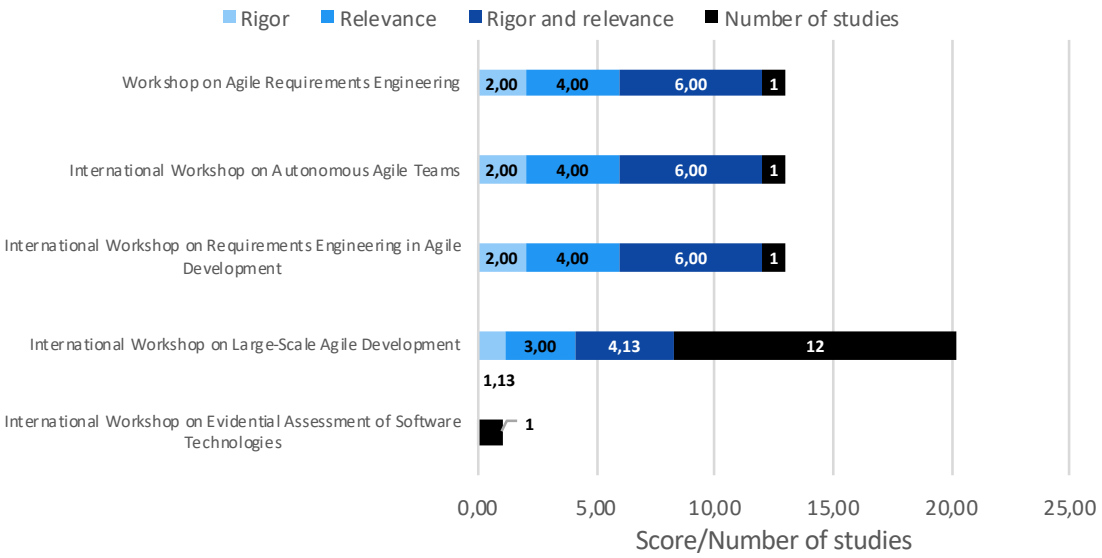


Figure 5.10.: Distribution of studies in workshops

per year has been published, compared to an average of 3.44 in the previous years. In 2019, three studies with the approach design and creation were published, whereas, in all previous years (2017 and 2008), only one study of this approach was published. Surveys (74.33 citations/year) are, on average, the most cited approach, followed by design and creation (54.4 citations/year) and SLR/SMS (39.33 citations/year). Finland (17 case studies), Sweden (13 case studies), and Norway (11 case studies) are the countries that have published the most case studies. Every second study using a grounded theory approach was published by scientists from the UK. Germany is the only country from which studies originate that use the approach theoretical. Action research, as well as design and creation, are the only approaches not used in all three publication channels, conferences, journals, and workshops. For case studies, 88.53% are of type evaluation research. A similar proportion of case studies can be calculated for the mixed methods (66.67%) and grounded theory (60%). The approaches philosophical papers, surveys, SLR/SMS, design and creation as well as action research use only one research type. Of the case studies, 85.25 percent are lessons learned. The four approaches theoretical, survey, SLR/SMS, and action research only provide lessons learned as research contributions. All approaches fell into the category of primary studies except SLR/SMS.

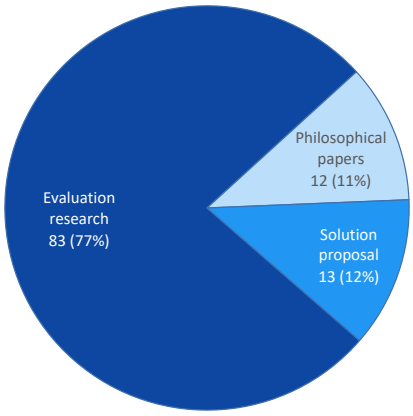


Figure 5.11.: Distribution of studies and research types

5.8. Research contributions of the studies

Research question 8: *What kinds of contributions are provided by studies related to large-scale agile development?*

Figure 5.13 shows the number of studies for the five contribution types. Most studies have the contribution type lessons leaned (78 studies), followed by the types model (10 studies), guidelines (9 studies), framework/methods (8 studies), and theory (3 studies). On average, 11.75 studies of the contribution type lessons learned were published in the last four years (2016-2019), whereas the average of the previous years (2008-2015) was significantly lower at 3.88. In 2019, four studies and thus, at least twice as many studies of the contribution type framework/methods were published as in all previous years (2018, 2017, 2014, and 2008). Most studies (3 studies) with the contribution type framework were published in 2019. The contribution type Framework/methods (∅ 48 citations/study) was cited most frequently, followed by the types Lessons learned (∅ 32.15 citations/study), Theory (∅ 26 citations/study) and Model (∅ 22.8 citations/study). Finland (18 studies), Sweden (15 studies), Norway (13 studies), and Germany (10 studies) have published 71.8% of the studies with the contribution type lessons learned. Germany has published the most studies (4 studies for both contribution types) with the contribution types guidelines and the frameworks. For the contribution type model, Norway and Germany share the first place for most publications with two studies each. All three studies of the contribution type theory are published by scientists from Norway, the UK, and Serbia. Across all contribution types, most studies were published in conferences. For the contribution type lessons learned, 52.56% of the studies were published in conferences, for the type model 77.78%, for

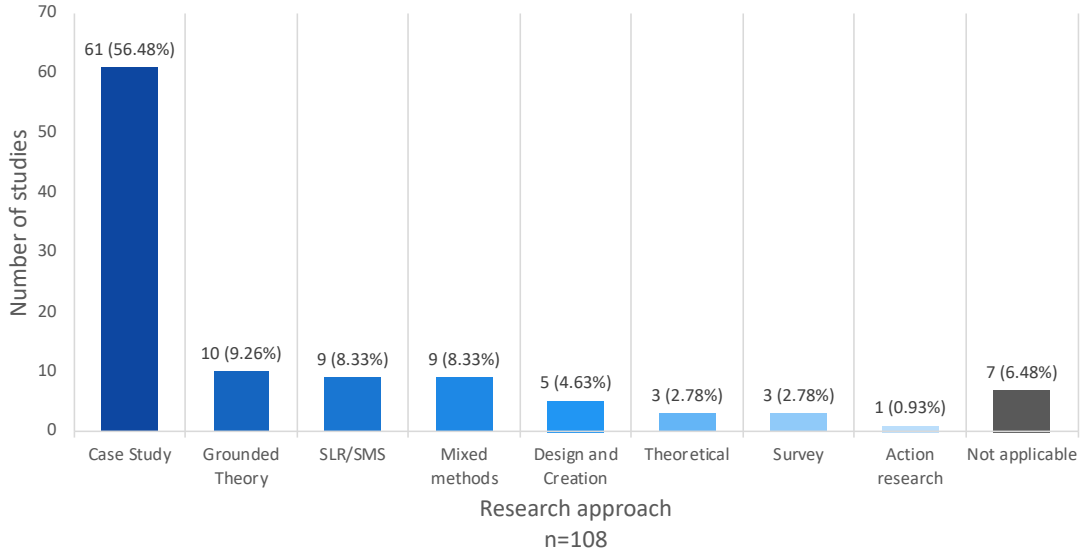


Figure 5.12.: Distribution of studies and research approaches

guidelines 44.44%, for framework/methods 62.5% and for theory 66.67%. Almost all of the studies (96.15%) with contribution type lessons learned belong to research type evaluation research. This also applies to 88.89% of the studies with the contribution type guidelines. Among the models, most studies (80%) are of the research type philosophical papers. The studies of the types framework/methods and theory are 100% and 66.67% Solution proposals. There are at least five times as many studies with the research approach case study as there are studies with any other approach that have lessons learned as contribution type. Secondary research occurs only in contribution type lessons learned (i.e., 11.54% of lessons learned studies). The studies with other contribution types are exclusively primary studies.

5.9. Primary and secondary research

Research question 9: *What is the proportion of primary and secondary research?*

Figure 5.14 shows the number of primary and secondary studies. Most studies are primary studies (91.67%). On average, 11.75 primary studies per year were published between 2014 and 2019, whereas the average in previous years (2008-2013) was significantly lower at 3.83. For the secondary studies, no trend regarding the publication numbers can be derived over the years. The average number of citations for secondary

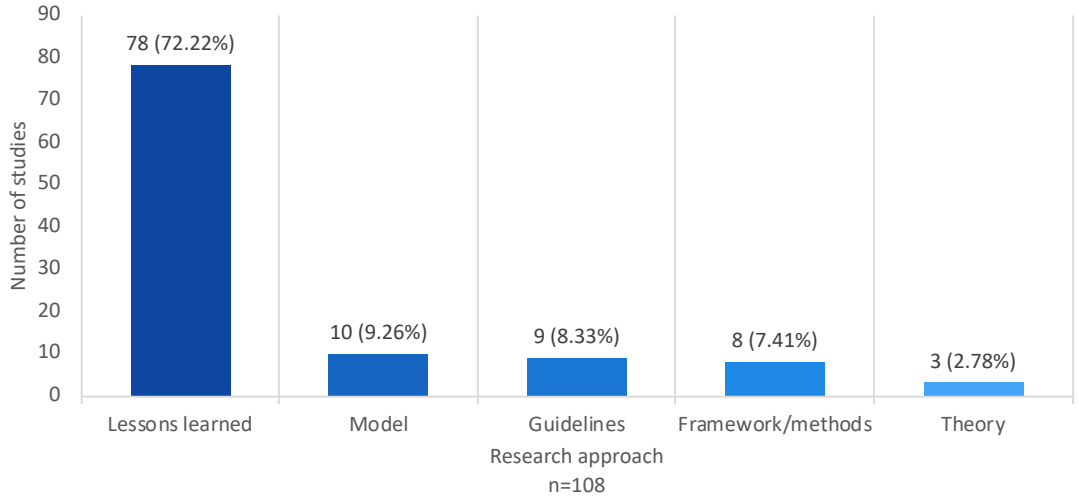


Figure 5.13.: Distribution of studies and research contributions

studies is 39.33 and thus above the average for primary studies (∅ 29.65 citations/study). Most of the primary studies were published by scientists from Finland (20 studies), Germany (18 studies), Sweden (17 studies), and Norway (16 studies). The secondary studies were mostly written by German (2 studies) and Indian (2 studies) scientists. Both types of research were most frequently published in conferences. However, the proportion of studies published in conferences is higher for primary studies (53.54%) than for secondary studies (44.44%). Most primary studies (74.75%) are of the type evaluation research. Secondary studies are exclusively of type evaluation research. There are at least six times as many primary studies that are case studies as primary studies with other research approaches. For secondary studies only the research approach SLR/SMS occurs. Within the 108 studies exist at least six times as many primary studies which have lessons learned as contribution type as primary studies with any other contribution type. Each of the secondary studies has the contribution type lessons learned.

5.10. Rigor and relevance of the studies

> ***Research question 10:*** *What is the rigor and relevance of the studies in large-scale agile development?*

Figure 5.15 gives an overview of the rigor and relevance assessments of the 99 primary studies. 91.92% of the primary studies were rated with the highest possible score for

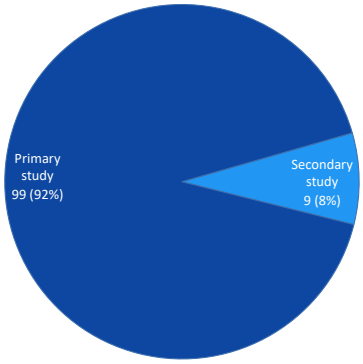


Figure 5.14.: Distribution of primary and secondary research

relevance (4 points/publication), and 68.69% have a rigor score of at least 2. Six primary studies received zero points for both rigor and relevance, which is the lowest possible rating. The average ratings for all primary studies are for rigor 2.1 points/publication relevance 3.68 points/publication and for rigor and relevance 5.77 points/publication. The rigor and relevance scores of the venues within conferences (Figure 5.8), journals (Figure 5.9) and workshops (Figure 5.10) are explained below. The highest scores for rigor were assigned to conference studies (∅ 2.22 points/publication), followed by journal studies (∅ 2.20 points/publication) and studies published in workshops (∅ 1.39 points/publication). In contrast to rigor, journal studies have the highest relevance scores (∅ 3.88 points/publication), ahead of conference studies (∅ 3.62 points/publication) and workshop publications (∅ 3.43 points/publication). Considering rigor and relevance, the studies published in journals have the highest average values (∅ 6.08 points/publication). The second-highest average values were assigned to conference studies (∅ 5.84 points/publication) and the lowest values to workshop studies (∅ 4.82 points/publication). The three journals Empirical Software Engineering, Project Management Journal, and IEEE Transactions on Software Engineering, have the highest average scores for relevance and rigor. Whereas five studies could be assigned to the venue Empirical Software Engineering, only one publication was assigned to the other two venues. Each study published in the journal IEEE Software has the top rating for rigor, but a score for relevance that is significantly below average. The journal IEEE Software has the highest number of studies within all journals but a rigor score below average. The conferences International Systems and Software Product Line Conference, as well as the International Conference on Information Systems, have the highest relevance and rigor scores compared to all other conferences. The

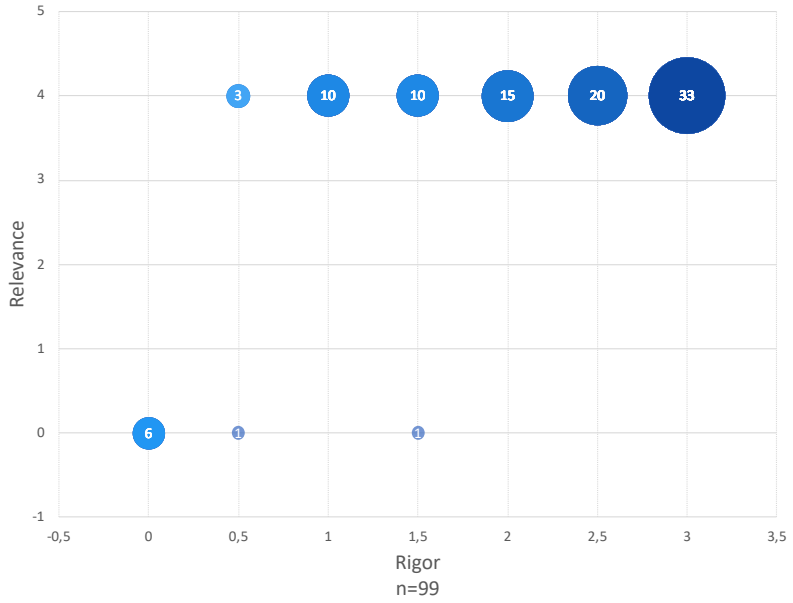


Figure 5.15.: Distribution of rigor and relevance scores

International Conference on Agile Software Development has the most publications among the conferences but just slightly scores above average concerning the relevance and rigor scores of the conferences. Four studies and thus third highest value in the number of publications were published within the Hawaii International Conference on System Sciences, which scored below average in all rigor and relevance scores. In the International Workshop on Large-Scale Agile Development, most studies were published across all workshops, but the values for relevance and rigor are below the average. In the following, the venues with the highest rigor and relevance scores and with at least three publications are presented for each of the three publication channels. Within the journals, the best-rated venue is Empirical Software Engineering, and under the conferences, the top venue is Euromicro Conference on Software Engineering. Across the workshops, the International Workshop on Large-Scale Agile Development is the leading venue. For the research types, the highest average values for rigor are obtained by evaluation studies (∅ 2.22 points/publication), followed by the studies of the type solution proposals (∅ 1.92 points/publication) and philosophical studies (∅ 1.54 points/publication). The highest average values for relevance is achieved by evaluation studies (∅ 3.89 points/publication), followed by the solution proposals (∅ 3.69 points/publication) and philosophical papers (∅ 2.33 points/publication). When ranking the types of the studies according to rigor and relevance evaluation papers (∅

6.11 points/publication) achieved the highest average values before the solution studies (∅ 5.62 points/publication) and philosophical papers (∅ 3.88 points/publication). Considering research approaches, the highest average values for rigor are achieved by grounded theory papers (∅ 2.40 points/publication), followed by mixed methods papers (∅ 2.39 points/publication) and case studies (∅ 2.33 points/publication). All approaches, except the category theoretical, have the highest possible relevance score (4.0 points/publication). Studies using the approach grounded theory (∅ 6.4 points/publication) achieved the highest score for relevance and rigor ahead of studies implementing mixed methods (∅ 6.39 points/publication) and case studies (∅ 6.33 points/publication). For the research contributions, the highest average values for rigor were achieved by papers contributing with lessons learned (∅ 2.25 points/publication), followed papers providing theory (∅ 2.00 points/publication) and papers delivering Framework/methods (∅ 1.94 points/publication). The highest average values for relevance was achieved by papers presenting theory (∅ 4.00 points/publication). The second highest relevance score on average is held by papers presenting Lessons learned (∅ 3.83 points/publication) ahead of papers developing Framework/methods (∅ 3.50 points/publication). For the sum of rigor and relevance, the highest averages were achieved for studies presenting lessons learned (∅ 6.07 points/publication), followed by theory studies (∅ 6.00 points/publication) and studies presenting Framework/methods (∅ 5.44 points/publication).

5.11. Research streams in large-scale agile development

> ***Research question 11:*** *Which research streams exist in large-scale agile development and which are the most active ones?*

Figure 5.24 shows which research streams exist with the help of a tree graph. An alternative visualization of the streams is provided in Table 5.1. The activity of the ten research streams, without subtopics, is analyzed in this section. In total, the selected studies could be assigned 131 times to the ten topics (see Figure 5.16). Most research studies were identified in stream Agile practices at scale (25 studies), Communication and coordination (24 studies), and Scaling agile frameworks (19 studies). In the streams Taxonomy (4 studies) and Agile portfolio management (2 studies), the least research studies were determined. In the following sections, the activity in the research streams is evaluated according to different dimensions.

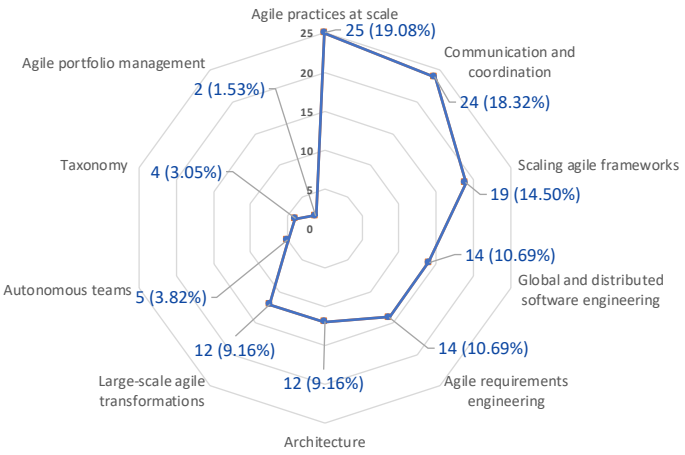


Figure 5.16.: Overview of activity in the research streams

5.11.1. Research streams, research types, and approaches

Figure 5.21 shows the relationship between research streams, research types, and approaches, as well as the number of studies. Evaluation research was implemented in the research streams 98 times (74.81%) and, therefore, five times more often than the second most frequently implemented type solution proposals (14.5%). In each stream, evaluation research is the most frequently used type. Philosophical papers (10.69%) hold the lowest proportion across the research streams. The type evaluation research was most frequently used in the stream Communication and coordination (19 times), followed by the streams Scaling agile frameworks (16 times) and Agile practices at scale (15 times). Within the stream Agile practices at scale (9 times), the type solution proposals was used at least three times as frequent as in the streams Scaling agile frameworks (3 times) and Architecture (2 times). For both streams Taxonomy (4 times) and Communication and coordination (4 times), the research type Philosophical papers was most frequently used, followed by the stream Agile requirements engineering (2 times). Across all streams, the approach case study was used 71 times. This frequency corresponds to five times of the absolute frequencies of the second-placed approaches Grounded theory and Mixed methods. In third place are SLR/SMS, which appear ten times in the streams. For all research streams except taxonomy, case studies are used most often.

5.11.2. Research streams and publication years

Figure 5.17 shows the relationship between research streams, publication years, and the number of studies. In the twelve years between 2008 and 2019, most studies were published in the streams Agile practices at scale (25 studies), Communication and coordination (24 studies), and Scaling agile frameworks (19 studies). Therefore, in total, 68 studies (51.91%) were published in the top three research streams. Overall, the least studies were published in the streams Agile portfolio management (2 studies), Taxonomy (4 studies), and Autonomous teams (5 studies). Over the last three years (2017-2019), most studies were published in the stream Communication and coordination followed by the streams Agile practices at scale (13 studies), Scaling agile frameworks (13 studies) as well as Architecture (10 studies). In the same period, the least number of publications were published within the streams Taxonomy (0 studies) and Agile portfolio management (1 study). The publication maximum was achieved in the streams Communication and coordination (2018) and Agile practices at scale (2019) with seven studies. Across all streams, an average of 23.67 studies was published between 2017 and 2019, significantly more than in the previous years, when, on average, 6.67 studies were published per year.

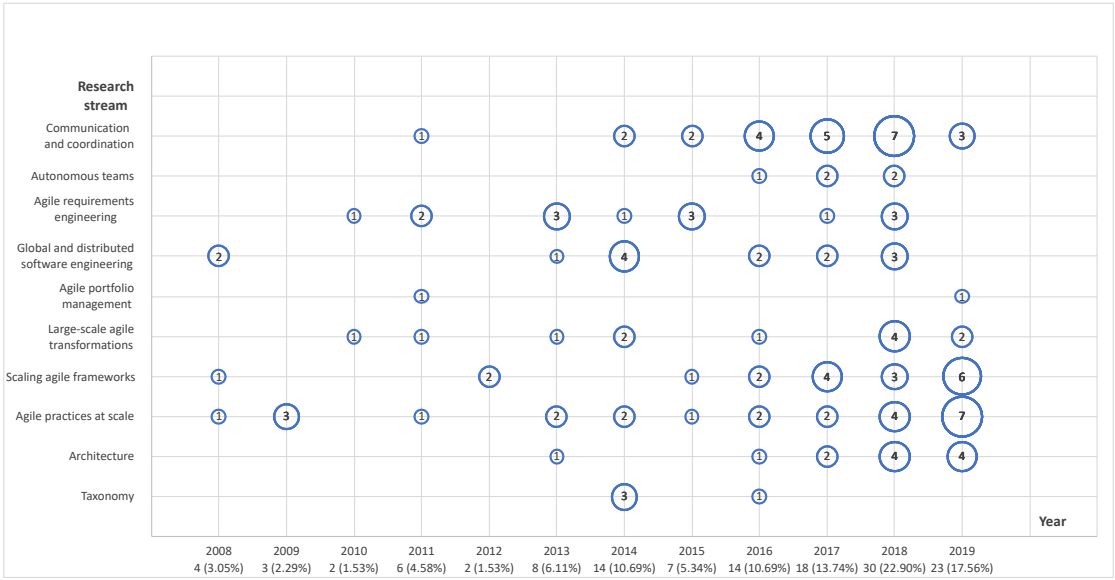


Figure 5.17.: Relationship between research streams, publication years, and the number of studies

5.11.3. Research streams and contributions

Figure 5.18 shows the relationship between research streams, contribution types, and the number of studies. Across all streams lessons learned was the most common contribution type with 70.23%. Lessons learned were more than seven times as frequent as framework and methods (9.92%) in second place. The type frameworks and methods occurs in the stream Agile practices at scale at least twice as often as in all other streams. Guidelines appear in the stream Scaling agile frameworks more than twice as often as in other streams. Agile practices at scale is the only stream in which all contribution types occur.

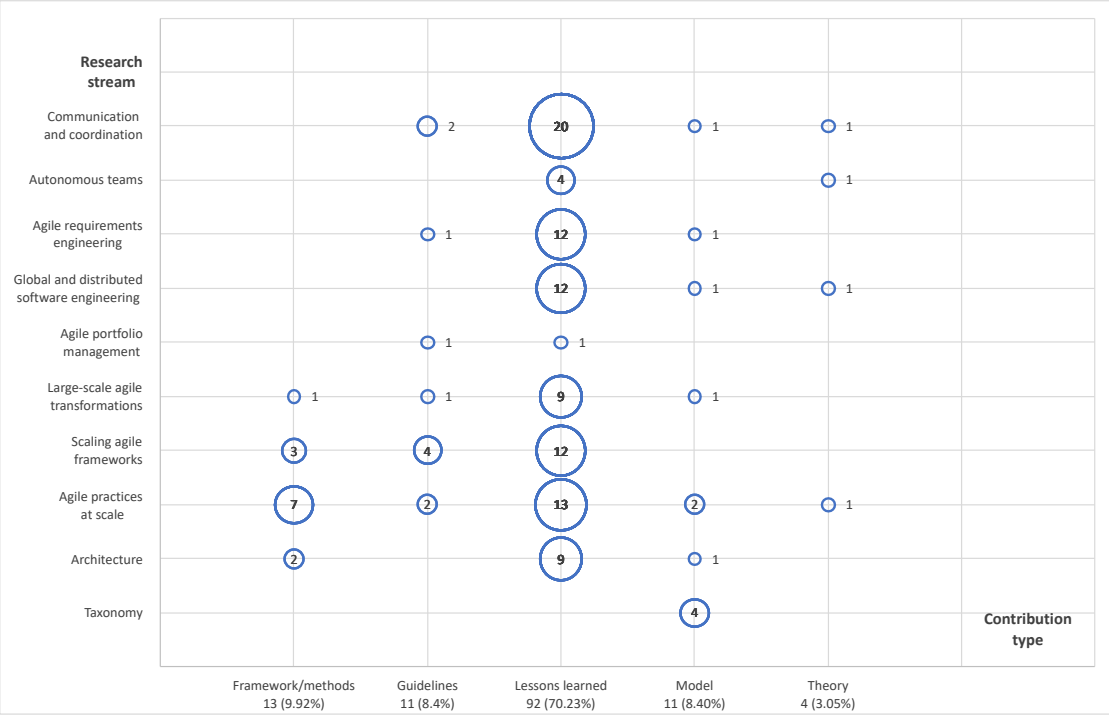


Figure 5.18.: Relationship between research streams, contribution types, and the number of studies

5.11.4. Research streams and primary and secondary studies

Figure 5.19 shows the relationship between research streams, primary and secondary research, as well as the number of studies. Primary research was used overall streams at least 12 times as often as secondary research. For each stream, primary studies were conducted more frequently than secondary studies. Most often, primary research occurs

in the streams Agile practices at scale (24 times), Communication and coordination (23 times), and Scaling agile frameworks (17 times). Secondary research occurs most frequently in the Global and distributed software engineering stream (3 times), where secondary research accounts for 27.27% of all studies, and thus exceeds the proportions of secondary studies in all other streams. There are no secondary studies in the streams Taxonomy, Agile portfolio management, Agile requirements engineering, and Autonomous teams. Especially in the stream Taxonomy, secondary research is suitable and would, therefore, be a promising future research topic.

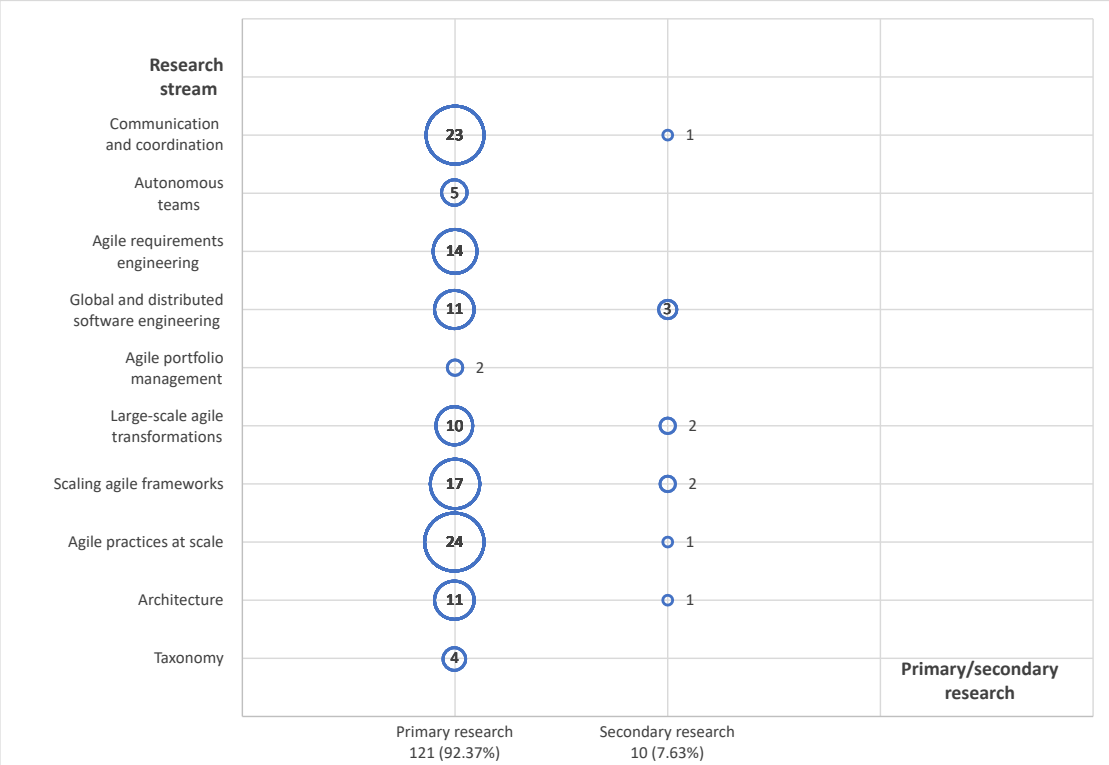


Figure 5.19.: Relationship between research streams as well as primary and secondary research

5.11.5. Research streams and publication channels

Figure 5.20 shows the relationship between research streams, publication channels and the number of studies. Overall, most studies were published in conferences. In all research streams the number of conference articles exceeds the publication numbers of other channels except in the stream Autonomous teams (1 study). In this

stream, more publications were published in workshops (2 studies) and journals (2 studies). Conferences articles appear most frequently in the streams Agile practices at scale (12), Scaling agile frameworks (11) and Architecture (9). The second most studies were published in the journals. Most journal studies fall on stream Agile practices at scale and the second largest number of studies on stream Communication and coordination. In the streams Agile portfolio management and Taxonomy, no journal articles exist. Most workshop articles have been published in the stream Communication and coordination. Within the streams Taxonomy, Large-scale agile transformations and Agile portfolio management no workshop articles appeared. The stream Communication and coordination has the smallest differences between the publication channels.

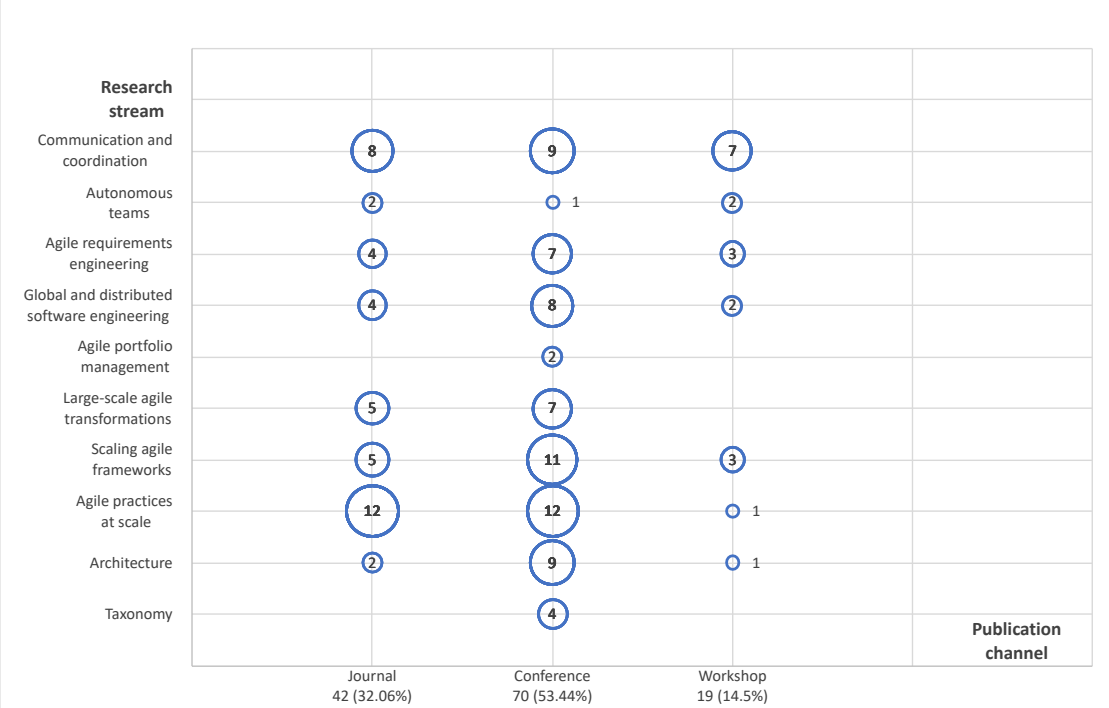


Figure 5.20.: Relationship between research streams, publication channels and the number of studies

5.11.6. Research streams and countries

Figure 5.22 shows the relationship between research streams, countries, and the number of studies. Overall, Germany (26 studies) is the most active country, followed by Sweden (24 studies), Finland (21 studies), and Norway (19 studies). The stream

Communication and coordination is led by Norway (10 studies) followed by Sweden (7 studies) and Germany (6 studies). In Norway, there is a strong focus on the stream Communication and coordination, in which more than five times as many studies were published compared to other streams. Sweden dominates the stream Agile requirements engineering with six studies before Finland (4 studies). The Global and distributed software engineering stream is led by UK (4 studies) ahead of India and Finland with three studies each. Finland has published in the stream Large-scale agile transformations with five studies more than twice as many as all other countries. Germany and Finland are both equally active in the stream Scaling agile frameworks with four studies each, followed by Sweden (3 studies). With ten countries involved, the Scaling agile frameworks stream is the stream in which most countries are involved. Most research in the stream Agile practices at scale is done in the countries of Germany and the UK with six studies each. Therefore, these countries published at least twice as many studies as in any other country. Germany dominates the stream Architecture since its researchers published at least times as much as all other countries have published on this topic. Within the other streams, Taxonomy and Autonomous teams, no clear leader country can be identified.

5.11.7. Research streams and citations per year

Figure 5.23 shows the citation numbers in the research streams in the period 2008 to 2019. Between 2015 and 2019, the average citation number per year was 269 across all streams, and thus below the average of the previous years (∅ 331.57 citations/year). Most citations are in stream Agile practices at scale (830 citations) followed by Scaling agile frameworks (560 citations) and Large-scale agile transformations (735 citations). Between 2017 and 2019, most citations occurred in the stream Communication and coordination (199 citations), followed by Architecture (111 citations), Agile requirements engineering (100 citations), and Scaling agile frameworks (100 citations). The maximum number of citations was reached in the stream Agile practices at scale (283 citations) in 2009. S23 is the study with the most citations (254 citations) and belongs to the stream Large-scale agile transformations.

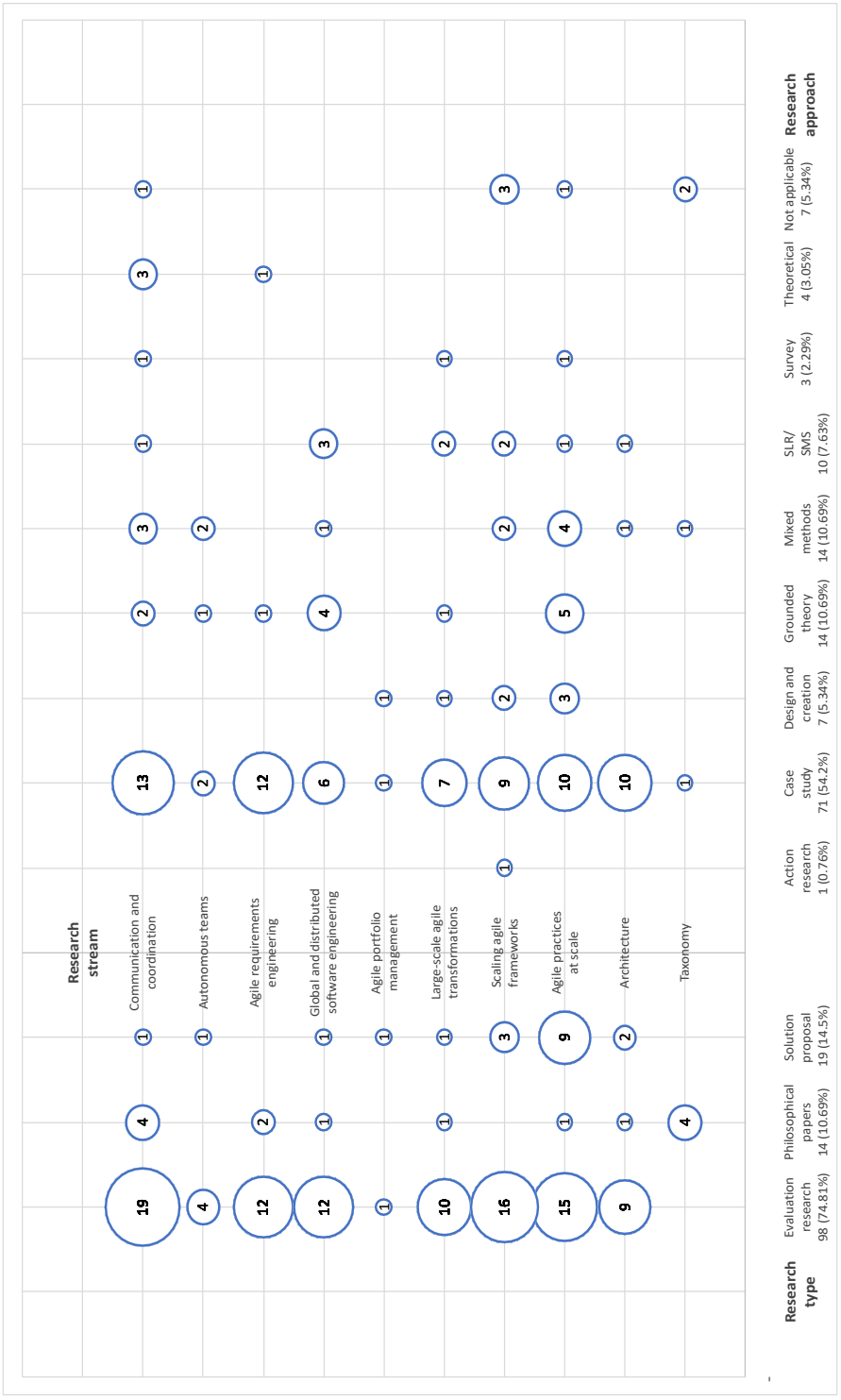


Figure 5.21.: Relationship between research streams, research types, and approaches

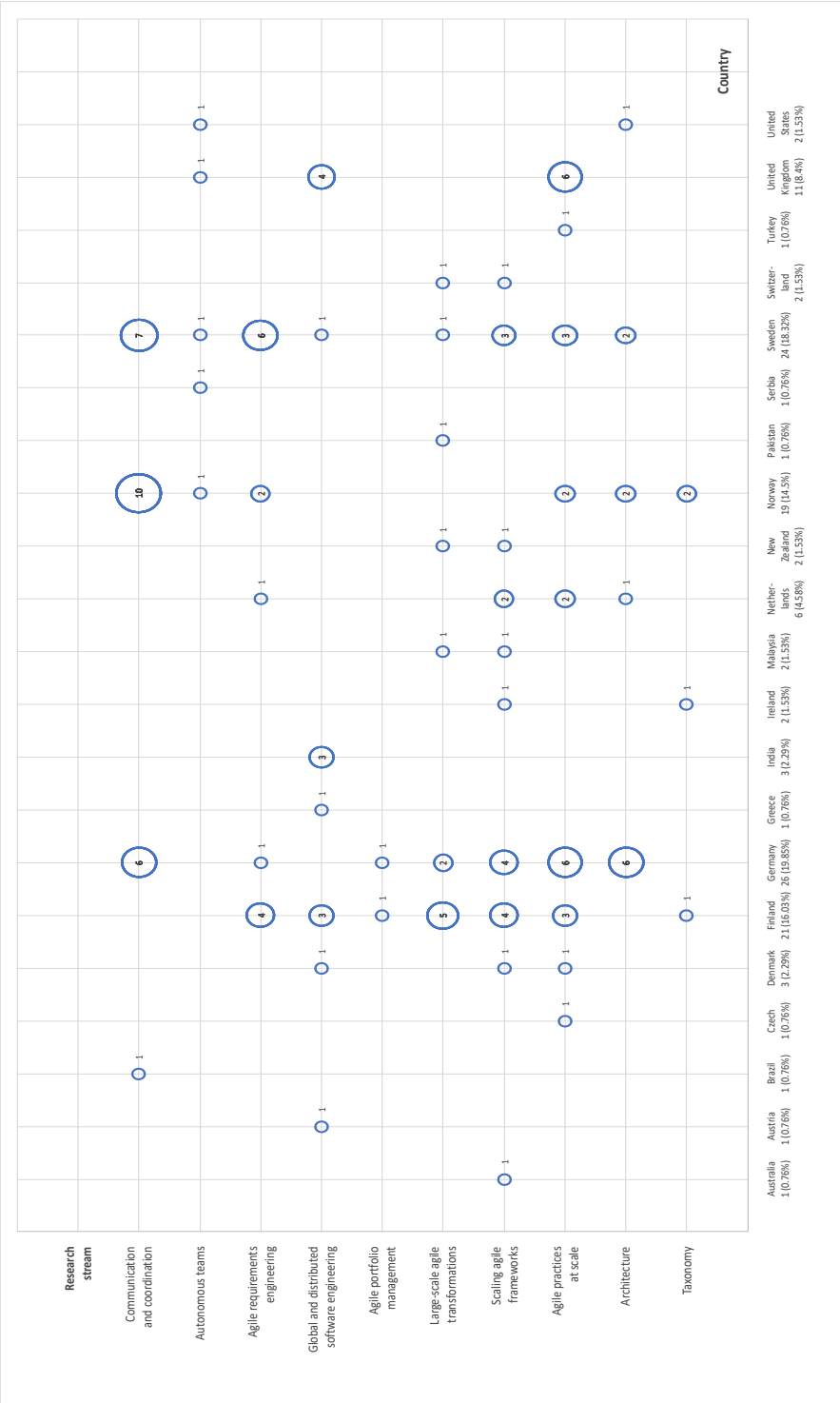


Figure 5.22.: Relationship between research streams, countries, and the number of studies

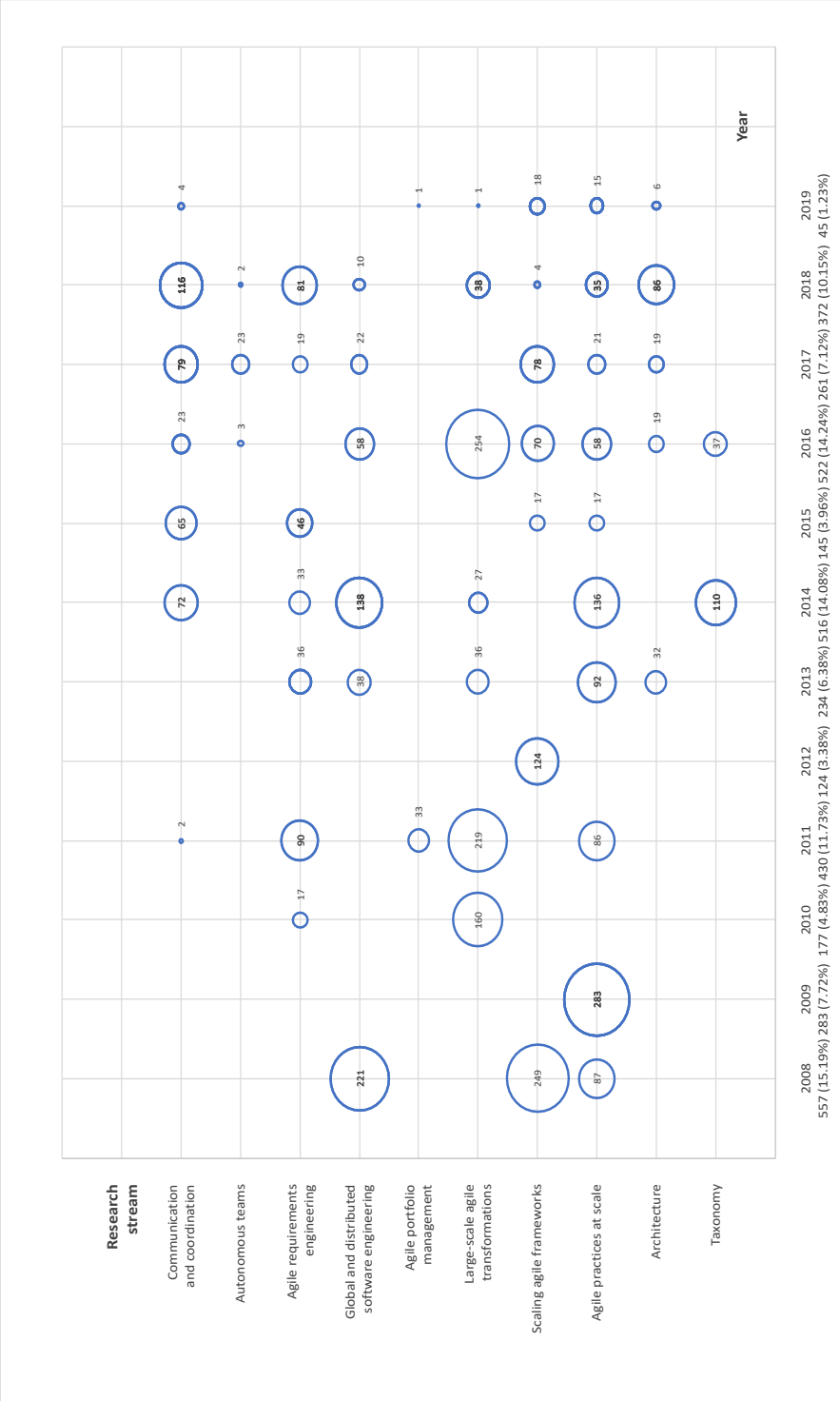


Figure 5.23.: Citation numbers in the research streams in the period 2008 to 2019

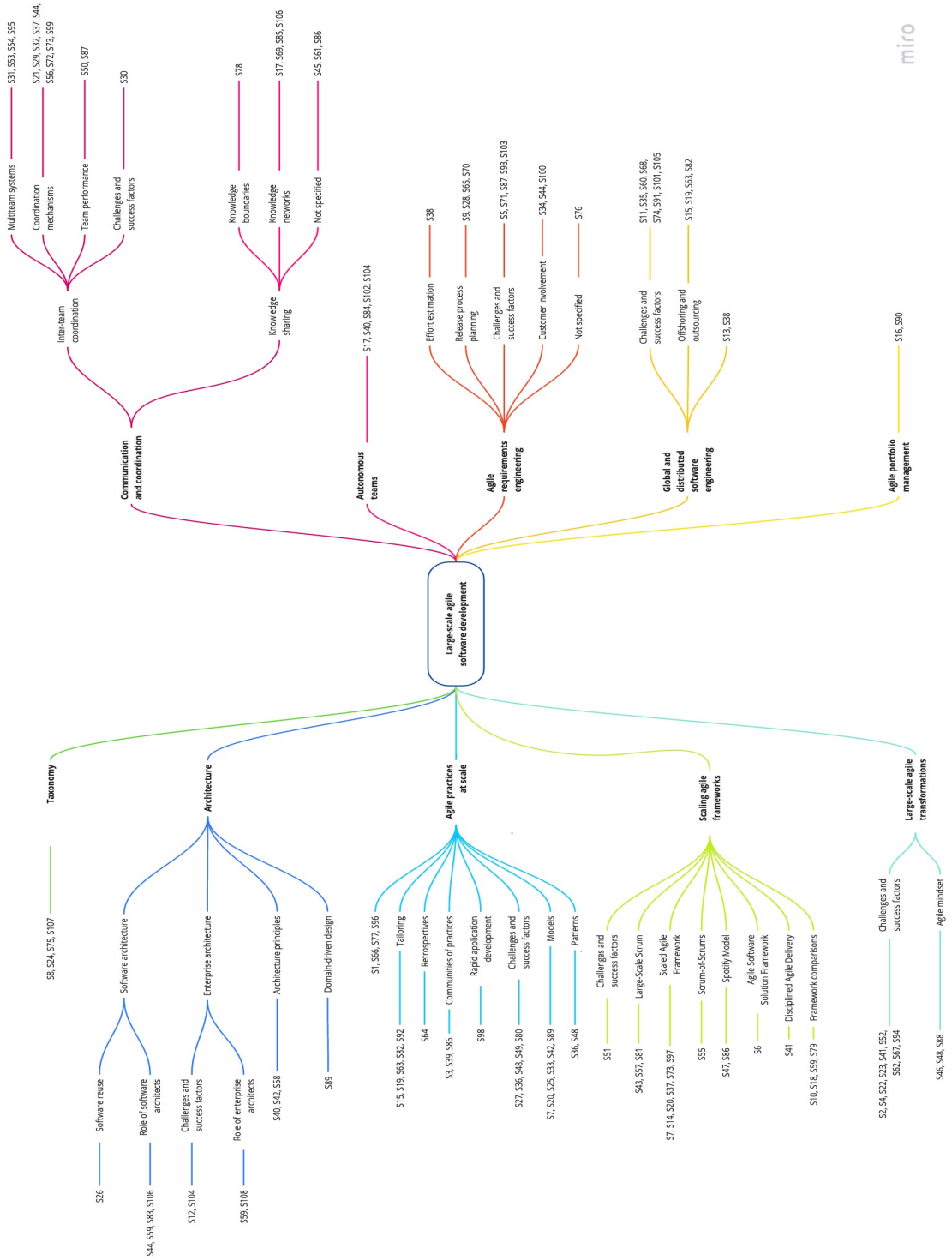


Figure 5.24.: Tree of the research streams

Topic	Subtopic	Sub-sub-topic	Sub-sub-sub-topic
Architecture	Software architecture	Software reuse	S26
		Role of software architects	S44, S59, S83, S106
	Enterprise architecture	Challenges and success factors	S12, S104
		Role of enterprise architects	S59, S108
	Architecture principles	S40, S42, S58	-
	Domain-driven design	S89	-
Communication and coordination	Knowledge sharing	Knowledge boundries	S78
		Knowledge networks	S17, S69, S85, S106
		Not specified	S45, S61, S86
	Inter-team coordination	Multiteam systems	S31, S53, S54, S95
		Coordination mechanisms	S21, S29, S32, S37, S44, S56, S72, S73, S99
		Team performance	S50, S87
		Challenges and success factors	S30
Autonomous teams	S17, S40, S84, S102, S104	-	-
Agile requirements engineering	Effort estimation	S38	-
	Release process planning	S9, S28, S65, S70	-
	Challenges and success factors	S5, S71, S87, S93, S103	-
	Customer involvement	S34, S44, S100	-
	Not specified	S76	-
Global distributed software development	Challenges and success factors	S11, S35, S60, S68, S74, S91, S101, S105	-
	Offshoring and outsourcing	S15, S19, S63, S82	-
	S13, S38		-
Agile portfolio management	S16, S90		-
Large-scale agile transformations	Challenges and success factors	S2, S4, S22, S23, S41, S52, S62, S67, S94	-
	Agile mindset	S46, S48, S88	-
Scaling agile frameworks	Challenges and success factors	S51	-
	Large-Scale Scrum	S43, S57, S81	-
	Sacled Agile Framework	S7, S14, S20, S37, S73, S97	-
	Scrum-of-Scrums	S55	-
	Spotify Model	S47, S86	-
	Agile Software Solution Framework	S6	-
	Disciplined Agile Delivery	S41	-
	Framework comparisons	S10, S18, S59, S79	-
Agile practices at scale	S1, S66, S77, S96	-	-
	Tailoring	S15, S19, S63, S82, S92	-
	Retrospectives	S64	-
	Communities of practices	S3, S39, S86	-
	Rapid application development	S98	-
	Challenges and success factors	S27, S36, S48, S49, S80	-
	Models	S7, S20, S25, S33, S42, S89	-
	Patterns	S36, S48	-
Taxonomy	S8, S24, S75, S107	-	-

Table 5.1.: The research streams in large-scale agile development

5.12. Research Agenda

Research question 12: *What are promising future research directions in the area of large-scale agile development?*

For each of the ten streams, the determined research questions are listed in Appendix A.3. In sum, 71 research questions were identified (see Figure 5.25). Most research questions were identified in stream Architecture (12 questions), Communication and coordination (12 questions), Large-scale agile transformations (10 questions), and Agile requirements engineering (10 questions). In the streams Taxonomy (1 questions) and Agile portfolio management (3 questions), the least research questions could be identified.

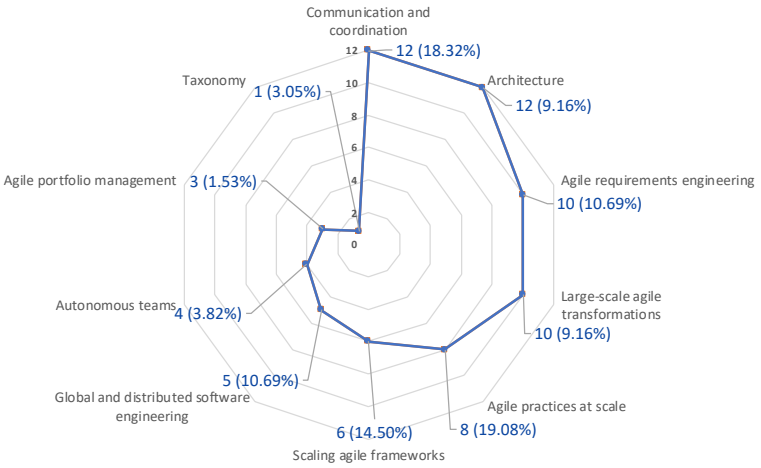


Figure 5.25.: Overview of identified research questions in research streams

6. Discussion

6.1. Key findings

Increasing attention of scientists for the field of large-scale agile development
The growing number of selected studies over the last twelve years shows that large-scale agile development is receiving increasing attention in science. Therefore, the scientific interest is simultaneously developing with the rising relevance of large-scale agile development in the industry [7]. In the last three years, German researchers published at least 60% more studies than all other countries. Every year most of the studies were of the type evaluation research, case study, and lessons learned. Therefore, very practice-oriented research was carried out in the field during all years.

Countries from Northern Europe have published the most
Most of the selected studies (84.3%) originate from Northern Europe. The country from which most publications stem is Finland (21 studies), followed by Germany (20 studies), Sweden (18 studies), and Norway (16 studies). The studies originating from these countries represent 69.44% of all selected studies. Only one selected study was published by researchers based in the United States, and Chinese researchers have not published any study relevant for this thesis.

S23 is the most salient study
The SLR (S23) by Dikert et al. [15] on challenges and success factors for Large-Scale Agile Transformations is the most salient study in the field. S23 has the highest total (254 citations) and average number (∅ 63.5 citations/year) of citations. Finland is the country with the highest average number of citations (among countries with at least three publications). Thus, it can be considered as the country from which most salient studies originate. Among the research types, the average citation numbers are similar. This means that the research type does not determine whether a study is salient. However, across the top 25 studies, 75% have the contribution type lessons learned.

Paasivaara and Lassenius are the most salient authors
Paasivaara and Lassenius are the most salient authors in the field of large-scale agile

development. They have the highest total number of publications and citations, as well as the best average citation numbers. Only in the case of average publication numbers, they do not occupy top positions. Here Uludağ and his co-authors Matthes and Kleehaus hold the first three ranks. The data shows that the years of experience are decisive for the number of citations, but not for the number of publications.

Between 2017 and 2019, the number of publications in journals grew faster than in conferences and workshop

In Conferences (57 studies), most of the papers were published. The number of journal papers (35 studies) sums up to the second-highest publication count, followed by workshop papers (16 studies). Since 2015, the number of journal publications is rising, whereas since 2016, the number of workshop publications is falling. Between 2017 and 2019, the number of publications in journals grew faster than in conferences and workshops. This indicates that the field is becoming more mature because journal papers generally have the highest maturity across all publication channels. In general, papers are published in top venues. Across the publication channels, the most notable venues are for conferences the International Conference on Global Software Engineering (11 studies) and the International Conference on Agile Software Development (10 studies). Under the journal venues, the highest number of studies belongs to IEEE Software (6 studies), Information and Software Technology (6 studies), and Empirical Software Engineering (5 studies). In the venue International Workshop on Large-Scale Agile Development (12 studies), most workshop papers were published. In the field of information systems research, so far, only a few publications have been published on the topic of large-scale agile development. This shortcoming of publications is exemplified by the International Conference on Information Systems (2 studies) Information Systems Journal (1 study).

Evaluation research is the predominant research type

Evaluation research is with 83 studies the predominant type within the selected studies. Only 13 studies are solution papers and 12 studies are philosophical papers. The research type had no effect on the average citation counts. An evaluation study was in 65.06% of the cases a case study and provided in 90.36% of the cases lessons learned. 66.67% of the philosophical papers provided models as contribution type and 61.54% of the solution papers resulted in framework/methods.

Most studies are case studies

The selected studies mainly consist of case studies (61 studies). Grounded theory (10 studies), SLR/SMS (9 studies), and mixed methods (9 studies) are the research

approaches that are behind case studies but do not even account together for the majority of publications. Between 2017 and 2019 on average ten case studies were published per year which outperforms the publication average of all previous years (∅ 3.44 publications/year). Surveys (74.33 citations/year) are, on average, the most cited approach, followed by design and creation (54.4 citations/year) and SLR/SMS (39.33 citations/year). Finland (17 case studies), Sweden (13 case studies), and Norway (11 case studies) are the countries that have published the most case studies. Dikert et al. [15] argued for the need to conduct case studies in the stream Large-scale agile transformations due to its high industry relevance. Whereas in 2016, Dikert et al. [15] only identified six papers investigating Large-scale agile transformations, this SMS assigned twelve studies to Large-scale agile transformations, and seven of the twelve studies are case studies.

Lessons learned are the most common contribution type
Most studies have the contribution type lessons learned (78 studies), followed by the types model (10 studies), and guidelines (9 studies). The number of studies providing lessons learned increased between 2016-2019 resulting in an average of 11.75 studies/year compared to the average in the previous years (3.88 studies/year). In 2019 the studies contributing framework/methods peaked as at least twice as many studies of the contribution type framework/methods were published as in all previous years (2018, 2017, 2014, and 2008). The contribution type Framework/methods (∅ 48 citations/study) was cited most frequently.

The field consists mostly of primary studies
Most studies are primary studies (91.67%). For the secondary studies, a trend cannot be derived regarding the publication numbers over the years. The average number of citations for secondary studies is 39.33 and thus above the average for primary studies (∅ 29.65 citations/study).

The primary studies have in general high relevance for industry
91.92% of the primary studies were rated with the highest possible score for relevance (4 points/publication), and 68.69% have a rigor score of at least two. The studies published in journals have the highest rigor and relevance scores (∅ 6.08 points/publication), ahead of conference studies (∅ 5.84 points/publication) and workshop studies (∅ 4.82 points/publication). Hereafter the venues with the highest rigor and relevance scores and with at least three publications are presented for each of the three publication channels. Within the journals the best rated venue is Empirical Software Engineering and under the conferences the top venue is Euromicro Conference on Software Engineering.

Across the workshops the International Workshop on Large-Scale Agile Development is the leading venue but has a score below the average for venues published in workshops.

Agile practices at scale and Communication and coordination are the most active research streams
Ten research streams and their corresponding subtopics were identified (see Figure 5.24). In 2016, Dikert et al. [15] criticized that little research has been conducted in the stream Agile practices at scale. In this SMS it has been determined that in the stream Agile practices at scale, the largest number of studies were published (25 studies), ahead of the stream Communication and coordination (24 studies). Additionally to the argumentation for the necessity to conduct more research in Agile practices at scale Dikert et al. [15] highlighted the shortcoming of papers that investigate which practices are used in companies and evaluate the related challenges and success factors. In this SMS, five papers (S27, S36, S48, S49, and S80) were identified which assess challenges and success factors of Agile practices at scale. Dikert et al. [15] mentioned that for Scaling frameworks, no studies exist that research the circumstances in which a particular framework is suitable. However in this SMS four studies were determined that consider framework comparisons (S10, S18, S59, and S79). Large-scale agile transformations only hold the 7th rank considering the number of studies but in the total number of citations, however, the third place. In the streams Agile portfolio management and Taxonomy, no journal articles exist. Therefore both streams are still at a low maturity level. Even if the stream Taxonomy is suitable for secondary research, no SLRs or SMSs were conducted. In consequence, secondary research would be a promising future research topic in this stream.

The streams Architecture and Communication and coordination offer the most research opportunities
RQs were identified for each of the ten research streams (see Figure 5.25). Most of the RQs were identified for the streams Architecture (12 RQs), Communication and coordination (12 RQs), Agile requirements engineering (10 RQs), and Large-scale agile transformations (10 RQs). Only one and, therefore, least RQs could be identified for the stream taxonomy. In none of the workshops that were examined for prospective research topics (Table A.2) have research questions on taxonomy been raised.

6.2. Limitations

This section describes the threats to validity and follows a discussion similar to Li

et al. [90]. Possibly, the results of this SMS are affected by the factors completeness of search, bias of study selection, and inexact data extraction. The results of PS and MS may not include all relevant studies in the area of large-scale agile development. The following actions were implemented to mitigate the risk of the incompleteness of the study search. First, the search for relevant literature followed a search strategy inspired by Zhang et al. [27] to compensate what the guidelines for SLRs lack and to achieve an appropriate search strategy design and execution. Second, only the most established search engines in the research community were used. Those search engines are indexing a large number of journals, conferences, and workshop publications. Third, within the PS, a manual and an automated search were implemented. The execution of a manual search had two objectives. The first goal was to ensure that all relevant publications, which were already known to the scientists in advance, would not be lost. The second goal was to increase the quality of relevant studies resulting from the preliminary search, as this set was used to assess the effectiveness of the search string belonging to the automated search. These actions do not guarantee the completeness of the study search, as the search term sets (see Section 4.1.4) may not contain all relevant terms. To mitigate the bias on study selection, two actions were performed. First, inclusion and exclusion criteria were defined and agreed upon to reach a common understanding of the characteristics that must be included in a relevant study. These criteria guided the filtering procedure. Second, the filtering procedure was performed by both scientists separately, and conflicts were resolved during discussions. In order to reduce the inaccuracy of data extraction, both scientists performed the data extraction process separately. Conflicts emerging from the keywording or from the assignment of items into the classification scheme were resolved by discussion.

7. Conclusion and future work

This chapter summarizes the thesis based on the research questions and presents an outlook for further work.

7.1. Summary

In this thesis, 108 studies were assessed as relevant to the field of large-scale agile development. The growing number of selected studies between 2008 and 2019 shows that large-scale agile development is receiving increasing attention from scientists. Most of the selected studies originate from Northern Europe, and Finland is the country where most studies were published. However, in the last three years, German researchers published more studies than all other countries. The most salient study in the field is a SLR by Dikert et al. [15], which determines challenges and success factors for Large-Scale Agile Transformations. Paasivaara and Lassenius are the most salient authors in the field of large-scale agile development because they have the highest total number of publications and citations,and the highest average of citations per year. The data shows that the years of experience are decisive for the number of citations, but not for the number of publications. Most of the studies were published in conferences. Nevertheless, since 2015, the number of journal publications is rising, and between 2017 and 2019, the number of publications in journals grew faster than in conferences and workshops. This indicates that the field is becoming more mature because journal papers generally have the highest maturity across all publication channels. Most of the studies were of the research type evaluation research, implemented a case study approach, provided lessons learned as contribution type, and could be classified as primary studies. In conclusion, very practice-oriented research was carried out in the field during the last twelve years. An assessment of the primary studies according to rigor and relevance showed that 91.92% the studies were rated with the highest possible score for relevance (4 points/publication), and 68.69% have a rigor score of at least two. Ten research streams and their corresponding subtopics were identified. Most studies were published in the stream Agile practices at scale, followed by Communication and coordination and Scaling agile frameworks. The stream Large-scale agile transformations only hold the 7th rank considering the number of studies but in the total number of citations, however, the third place. In the streams Agile portfolio management and

Taxonomy, no journal articles exist. Therefore both streams are still at a low maturity level. Even if the stream Taxonomy is suitable for secondary research, no SLRs or SMSs were conducted. In consequence, secondary research would be a promising future research topic in this stream. RQs were identified for each of the ten research streams. Most of the RQs were identified for the streams Architecture, Communication and coordination, Agile requirements engineering, and Large-scale agile transformations.

7.2. Future work

The following recommendations for future work are based on the identified research questions in this SMS (see Section 5.12). Within the research stream Communication and coordination, it would be a promising future research direction to investigate how coordination mechanisms can be effectively applied in large-scale agile development. Further, it would be interesting to determine how effective knowledge networks can be created. With regard to autonomous teams, research projects could be conducted in order to identify factors that enable team autonomy. Besides, balancing inter-team coordination and team autonomy in agile projects could be another aspect of future work. Considering Global distributed software development, identification of challenges, benefits, and success factors of applying agile practices in distributed projects seems to provide many opportunities for future research projects. Within the stream, Large-scale agile transformations the identification of reasons and consequences of conducting large-scale agile transformations would be a suitable research endeavor. Moreover, researchers of the field could be interested in investigating the factors that enable the integration of non-agile units with agile organizational units to support agile transformations. The stream Scaling agile frameworks also offers research opportunities. For example, it would be essential to increase the knowledge on the issue of effectively selecting scaling agile frameworks that are suitable for specific contexts. The last suggestion for future research refers to Agile practices at scale. Conducting research on the challenges, benefits, and success factors of scaling agile practices in organizations is a research direction considered necessary by multiple scientists and practitioners.

A. Appendix

A.1. Selected studies

ID / Citation	Title
S1 / [91]	A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case
S2 / [92]	Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation
S3 / [12]	Communities of practice in a large distributed agile software development organization - Case Ericsson
S4 / [93]	"Agile meets non-agile": Implications of adopting agile practices at enterprises
S5 / [94]	A case study on benefits and side-effects of agile practices in large-scale requirements engineering
S6 / [95]	A framework to support the evaluation, adoption and improvement of agile methods in practice
S7 / [96]	A Maturity Model for Scaling Agile Development
S8 / [97]	A Model for Understanding When Scaling Agile Is Appropriate in Large Organizations
S9 / [98]	A Revelatory Case Study on Scaling Agile Release Planning
S10 / [5]	A Review of Scaling Agile Methods in Large Software Development
S11 / [99]	A systematic literature review to identify human related challenges in globally distributed agile software development: towards a hypothetical model for scaling agile methodologies
S12 / [100]	Adaptation of enterprise architecture efforts to an agile environment
S13 / [101]	Adapting to Changes in a Project's DNA: A Descriptive Case Study on the Effects of Transforming Agile Single-Site to Distributed Software Development
S14 / [102]	Adopting SAFe to Scale Agile in a Globally Distributed Organization

Table A.1 continued from previous page

ID / Citation	Title
S15 / [103]	Agile Method Tailoring in Distributed Enterprises: Product Owner Teams
S16 / [104]	Agile Portfolio Management: Design Goal and Principles
S17 / [105]	Agile Teams in Large-Scale Distributed Context: Isolated or Connected?
S18 / [106]	Comparing Scaling Agile Frameworks Based on Underlying Practices
S19 / [107]	Artefacts and agile method tailoring in large-scale offshore software development programmes
S20 / [108]	Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework
S21 / [109]	Assigned roles for inter-team coordination in large-scale agile development: a literature review
S22 / [110]	Becoming agile in the digital transformation: The process of a large-scale agile transformation
S23 / [15]	Challenges and success factors for large-scale agile transformations: A systematic literature review
S24 / [6]	Characteristics and Principles of Scaled Agile
S25 / [111]	Combining agile software projects and large-scale organizational agility
S26 / [112]	Communication factors for speed and reuse in large-scale agile software development
S27 / [113]	Complex software project development: agile methods adoption
S28 / [114]	Continuous Release Planning in a Large-Scale Scrum Development Organization at Ericsson
S29 / [115]	Coordinating knowledge work in multiteam programs: findings from a large-scale agile development program
S30 / [116]	Coordination Challenges in Large-Scale Software Development: A Case Study of Planning Misalignment in Hybrid Settings
S31 / [10]	Coordination in Large-Scale Agile Software Development: A Multiteam Systems Perspective
S32 / [117]	Coordination in multi-team programmes: An investigation of the group mode in large-scale agile software development
S33 / [118]	Crossing the Boundaries - Agile Methods in Large-Scale, Plan-Driven Organizations: A Case Study from the Financial Services Industry

Table A.1 continued from previous page

ID / Citation	Title
S34 / [119]	Customer-Specific Teams for Agile Evolution of Large-Scale Embedded Systems
S35 / [120]	Distributed Agile Development: Using Scrum in a Large Project
S36 / [121]	Documenting Recurring Concerns and Patterns in Large-Scale Agile Development
S37 / [122]	Dynamics of Inter-Team Coordination Routines in Large-Scale Agile Software Development
S38 / [123]	Effort estimation in large-scale software development: An industrial case study
S39 / [124]	Empower Your Agile Organization: Community-Based Decision Making in Large-Scale Agile Development at Ericsson
S40 / [125]	Enabling autonomous teams in large-scale agile through architectural principles
S41 / [126]	Enhancing Product and Service Capability Through Scaling Agility in a Global Software Vendor Environment
S42 / [127]	Establishing Architecture Guidelines in Large-Scale Agile Development Through Institutional Pressures
S43 / [128]	Experiences in scaling the Product Owner role in large-scale globally distributed Scrum
S44 / [19]	Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation
S45 / [129]	Fostering effective inter-team knowledge sharing in agile software development
S46 / [130]	Hidden facilitators of agile transition: Agile coaches and agile champions
S47 / [131]	How Enterprises Adopt Agile Structures: A Multiple-Case Study
S48 / []	Identifying and Documenting Recurring Concerns and Best Practices of Agile Coaches and Scrum Masters in Large-Scale Agile Development
S49 / [132]	Identifying and Structuring Challenges in Large-Scale Agile Development Based on a Structured Literature Review
S50 / [133]	Impacts on team performance in large-scale agile software development

Table A.1 continued from previous page

ID / Citation	Title
S51 / [134]	Implementing Large-Scale Agile Frameworks: Challenges and Recommendations
S52 / [135]	Integrating global sites into the lean and agile transformation at ericsson
S53 / [136]	Inter-Team Coordination in Large Agile Software Development Settings: Five Ways of Practicing Agile at Scale
S54 / [137]	Inter-team coordination in large-scale agile development: A case study of three enabling mechanisms
S55 / [138]	Inter-team Coordination in Large-scale Globally Distributed Scrum: Do Scrum-of-scrums Really Work?
S56 / [139]	Inter-team coordination mechanisms in large-scale agile
S57 / [140]	Investigating the Adoption and Application of Large-Scale Scrum at a German Automobile Manufacturer
S58 / [141]	Investigating the Establishment of Architecture Principles for Supporting Large-Scale Agile Transformations
S59 / [21]	Investigating the Role of Architects in Scaling Agile Frameworks
S60 / [142]	Is Scrum Fit for Global Software Engineering?
S61 / [143]	Knowledge Sharing and Process Improvement in Large-Scale Agile Development
S62 / [144]	Large-scale agile transformation at Ericsson: a case study
S63 / [145]	Large-Scale Offshore Agile Tailoring: Exploring Product and Service Organisations
S64 / [146]	Learning in the large - an exploratory study of retrospectives in large-scale agile development
S65 / [147]	Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson
S66 / [148]	Migrating Defect Management from Waterfall to Agile Software Development in a Large-Scale Multi-site Organization: A Case Study
S67 / [64]	Motivators for Large-Scale Agile Adoption From Management Perspective: A Systematic Literature Review
S68 / [149]	Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects.
S69 / [150]	Networking in a large-scale distributed agile project

Table A.1 continued from previous page

ID / Citation	Title
S70 / [151]	Operational release planning in large-scale Scrum with multiple stakeholders - A longitudinal case study at F-Secure Corporation
S71 / [152]	Perspectives on Productivity and Delays in Large-Scale Agile Projects
S72 / [153]	Planned and unplanned meetings in large-scale projects
S73 / [154]	Practices for Vertical and Horizontal Coordination in the Scaled Agile Framework
S74 / [155]	Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process
S75 / [20]	Problematizing agile in the large: Alternative assumptions for large-scale agile development
S76 / [156]	Re-conceptualizing requirements engineering: findings from a large-scale, agile project
S77 / [157]	Relationships between Project Size, Agile Practices, and Successful Software Development: Results and Analysis
S78 / [158]	Scaling Across Knowledge Boundaries: A Case Study Of A Large-Scale Agile Software Development Project
S79 / [159]	Scaling agile: how to select the most appropriate framework
S80 / [160]	Scaling agile in large organizations: Practices, challenges, and success factors
S81 / [161]	Scaling Scrum in a Large Globally Distributed Organization: A Case Study
S82 / [162]	Scrum Master Activities: Process Tailoring in Large Enterprise Projects
S83 / [163]	Software Architects in Large-Scale Distributed Projects: An Ericsson Case Study
S84 / [164]	Software Development and CSCW: Standardization and Flexibility in Large-Scale Agile Development
S85 / [13]	Software teams and their knowledge networks in large-scale software development
S86 / [165]	Spotify Guilds: How to Succeed with Knowledge Sharing in Large-Scale Agile Organizations
S87 / [166]	Steering through Incentives in Large-Scale Lean Software Development

Table A.1 continued from previous page

ID / Citation	Title
S88 / [167]	Supporting a Large-Scale Lean and Agile Transformation by Defining Common Values
S89 / [168]	Supporting large-scale agile development with domain-driven design
S90 / [169]	Supporting Scaling Agile with Portfolio Management: Case Paf.com
S91 / [170]	Systematic Review of Success Factors for Scaling Agile Methods in Global Software Development Environment: A Client-Vendor Perspective
S92 / [171]	Tailoring Product Ownership in Large-Scale Agile Projects: Managing Scale, Distance, and Governance
S93 / [172]	Technical Dependency Challenges in Large-Scale Agile Software Development
S94 / [17]	The effect of moving from a plan-driven to an incremental software development approach with agile practices: An industrial case study
S95 / [173]	The Effects of Team Backlog Dependencies on Agile Multiteam Systems: A Graph Theoretical Approach
S96 / [174]	The Impact of Agile Principles and Practices on Large-Scale Software Development Projects: A Multiple-Case Study of Two Projects at Ericsson
S97 / [175]	The SAFe Way to the Agile Organization
S98 / [176]	The utility of rapid application development in large-scale, complex projects
S99 / [177]	To schedule or not to schedule? An investigation of meetings as an inter-team coordination mechanism in large-scale agile software development
S100 / [178]	Towards continuous validation of customer value
S101 / [179]	Towards Rapid Releases in Large-Scale XaaS Development at Ericsson: A Case Study
S102 / [180]	Transition of organizational roles in Agile transformation process: A grounded theory approach
S103 / [181]	Understanding Challenging Situations in Agile Quality Requirements Engineering and Their Solution strategies: Insights from a Case Study
S104 / [182]	Unpacking agile enterprise architecture innovation work practices: A qualitative case study of a railroad company

Table A.1 continued from previous page

ID / Citation	Title
S105 / [183]	Using scrum in a globally distributed project: A case study
S106 / [184]	Using social network analysis to investigate the collaboration between architects and agile teams: A case study of a large-scale agile development program in a german consumer electronics company
S107 / [56]	What is large in large-scale? a taxonomy of scale for agile software development
S108 / [185]	What to Expect from Enterprise Architects in Large-Scale Agile Development? A Multiple-Case Study

Table A.1.: Selected studies

A.2. Selected workshops

ID / Citation	Title
W1 / [186]	1st International Workshop on Large-Scale Agile Development
W2 / [187]	2nd International Workshop on Large-Scale Agile Development
W3 / [188]	4th International Workshop on Large-Scale Agile Development
W4 / [189]	5th International Workshop on Large-Scale Agile Development
W5 / [190]	6th International Workshop on Large-Scale Agile Development
W6 / [191]	7th International Workshop on Large-Scale Agile Development
W7 / [192]	1st International Workshop on Agile Transformation
W8 / [193]	1st International Workshop on Autonomous Teams
W9 / [194]	2nd International Workshop on Autonomous Teams

Table A.2.: Selected workshops

A.3. Research agenda

Architecture

1. How is the role of enterprise architects practiced in large-scale agile development? [W5, S12, S108]
2. How do architects collaborate with agile teams in large-scale agile development? [W5, S83, S89]
3. How can coordination mechanisms improve architecture sharing at intra- and inter-team level? [S44, S106]
4. How can architecture drive large-scale agile transformations? [W1, W3]
5. How can technical debts be minimized in large-scale agile projects? [W2, W6]
6. How can the decision-making power between architects and agile teams be balanced? [W5, S106]
7. How can software architecture support the coordination of agile teams? [W6, W8]
8. How can emergent and intentional architecture be balanced? [W5]
9. Which typical challenges do architects face in large-scale agile development? [S59]
10. What is the effect of applying architecture principles on the outcome of large-scale agile transformations? [S58]
11. What are good practices for addressing challenges related to the establishment of architecture principles? [S58]
12. How can the compliance of agile teams with architecture principles automatically determined? [S42]

Agile practices at scale

1. What are challenges, benefits, and success factors of scaling agile practices in organizations? [W1, W2, W3, S2, S23, S80]
2. What are recurring concerns and good practices of typical stakeholders in large-scale agile development? [S1, S36, S48, S80, S89]
3. How can the onboarding of new agile team members be facilitated in large-scale agile projects? [W3, W6, W9]
4. What are challenges, benefits, and success factors of establishing communities of practice in large-scale agile projects? [W1, W3, W6, S3]
5. What are appropriate metrics to monitor the progress of agile teams and to support transparency in large-scale agile projects? [W2]
6. What is the impact of applying agile practices to the overall performance of the organization? [S65]
7. How can agile practices be scaled in organizations from the public sector? [W6]
8. How can continuous improvement at intra- and inter-team level be facilitated? [S61]

9. Which issues arise when are retrospectives are organized at inter-team level? [S64]

Large-scale agile transformations

1. What are challenges, benefits, and success factors of performing large-scale agile transformations? [W4, W6, W7, S4, S22, S23, S25, S47, S62, S65, S101, S102]
2. What are reasons and consequences of conducting large-scale agile transformations on the organization? [S5, S5, S25, S47, S67]
3. How can non-agile units be integrated with agile organizational units to support agile transformations? [W7, S4, S16, S23, S47]
4. How are agile structures adopted in business units that are not engaged in IT development or delivery? [W4, W6, S47]
5. How can hierarchical and organizational structures be reduced to facilitate large-scale agile transformation? [W4, W6, W7]
6. How can local optimization of agile teams be aligned with the enterprise strategy? [W6, S87]
7. How do agile teams adopt common values within large-scale agile transformations? [S88]
8. Which KPIs exist to measure the enterprise agility? [W6]
9. How are agile methods adopted at large-scale in highly regulated environments? [S33]
10. What are the responsibilities of agile coaches in large-scale agile transformation? [S46]

Taxonomy

1. How can agile in the large be conceptualized besides the dimension of number of teams? [S78, S107]

Scaling agile frameworks

1. How can scaling agile frameworks be selected that are suitable for specific contexts? [W3, S14, S23, 62, S81, S94]
2. Which scaling agile frameworks are used in organizations and what are their benefits and challenges? [W5, W6, S14, S23, S41, S62, S81]
3. How is the Scaled Agile Framework (SAFe) adopted in organizations and what are respected challenges and risks when adopting it? [W5, S7, S20]
4. How are scaling agile frameworks tailored to meet the needs of the organizations in which they are adopted? [S23, S62]

5. How and when should be scaling agile frameworks used in large-scale agile projects? [W4, S10]
6. How is Large-Scale Scrum adopted in different types of organizations? [S57]

Agile portfolio management

1. What are best practices in agile portfolio management? [W6, S16]
2. How is portfolio management interrelated with other governance functions in an agile context? [S16]
3. How can traditional portfolio management techniques be applied in an agile environment? [S90]

Global and distributed software development

1. What are challenges, benefits, and success factors of applying agile practices in distributed projects? [W1, S11, S13, S14, S35, S74, S81, S105]
2. Which human related factors can positively affect globally distributed software projects? [S11]
3. How is frequent communication in distributed projects enabled to overcome the challenges of distance? [S105]
4. How is knowledge sharing performed in distributed large-scale agile projects? [W6]
5. How can virtual agile teams be supported in distributed software development projects? [W3]

Agile requirements engineering

1. How do organizations that have adopted agile methods implement release planning? [W1, S9, S28]
2. How do product owners and customers collaborate with developers in large-scale agile projects? [W1, W2]
3. What legal limitations exist in contracts that reduce agility in large scale projects? [W1, W2]
4. What are good contracting models for organizations with external customers? [W6]
5. What are typical requirements engineering challenges in large-scale agile development? [W7]
6. How can the prioritization between functional and non-functional requirements be balanced in large-scale agile projects? [W2]
7. How can technical dependencies between agile teams be minimized? [S93]

8. What are factors that impact the accuracy of effort estimations in large scale agile projects? [S38]
9. How can customer representatives and agile teams be aligned in large-scale agile projects? [S44]
10. How can high-level planning elements be incorporated in agile daily routines of large-scale agile projects? [S53]

Autonomous teams

1. How can team autonomy in large-scale agile development be increased? [W4, W6, W8, W9, S78S8, S17, S30, S40]
2. How can inter-team coordination and team autonomy be balanced in large-scale agile projects? [W8, W9, S17, S44]
3. What are the effective intra- and inter-team coordination mechanisms for autonomous agile teams? [W8, W9]
4. How can autonomous teams be designed, supported, and coached? [W8]

Communication and coordination

1. How can coordination mechanisms be applied effectively in large-scale agile development? [W1, W2, W4, W6, W7, S17, S19, S29, S37, S55, S95]
2. How can effective knowledge networks be created in large-scale agile projects? [W1, W2, W3, W6, S45]
3. Which tools can be used to support inter-team coordination in large-scale agile projects? [W5, W6, W8, S72]
4. Based from a multiteam perspective, how is coordination in large-scale agile development performed? [S31, S54]
5. How is intra-team coordination affected by increased focus on inter-team coordination or vice versa? [S29, S73]
6. How can the number of meetings in large-scale agile projects be reduced? [W5, S72]
7. How can meetings in large-scale projects be designed in order to increase the effectiveness of coordination? [W6, S72]
8. What are effective organizational structures and collaboration models in large projects? [W1]
9. Which effect do cultural differences have on inter-team coordination large-scale agile projects? [W6]
10. How can daily stand-up meetings be organized in a way that they enable inter-team coordination? [S56]
11. How can focused work and knowledge be balanced in large-scale agile projects?

[W6]

12. How does co-location of agile teams affect knowledge sharing in large-scale agile projects? [W6]

A.4. MS search terms

Search engine	Search string
DB1	((("Document Title":"large scale" OR "Abstract":"large scale"),OR ("Document Title":"scaling" OR "Abstract":"scaling")) AND (("Document Title":"agile" OR "Abstract":"agile") OR ("Document Title":"agility" OR "Abstract":"agility") OR ("Document Title":"extreme programming" OR "Abstract":"extreme programming") OR ("Document Title":"XP" OR "Abstract":"XP") OR ("Document Title":"feature driven development" OR "Abstract":"feature driven development") OR ("Document Title":"FDD" OR "Abstract":"FDD") OR ("Document Title":"scrum" OR "Abstract":"scrum") OR ("Document Title":"crystal" OR "Abstract":"crystal") OR ("Document Title":"pair programming" OR "Abstract":"pair programming") OR ("Document Title":"test-driven development" OR "Abstract":"test-driven development") OR ("Document Title":"TDD" OR "Abstract":"TDD") OR ("Document Title":"leanness" OR "Abstract":"leanness") OR ("Document Title":"lean software development" OR "Abstract":"lean software development") OR ("Document Title":"lean development" OR "Abstract":"lean development") OR ("Document Title":"LSD" OR "Abstract":"LSD")) AND NOT ("Document Title":"manufacturing" OR "Abstract":"manufacturing")) OR (("Document Title":"Crystal Family" OR "Abstract":"Crystal Family") OR ("Document Title":"Dynamic Systems Development Method Agile Project Framework for Scrum" OR "Abstract":"Dynamic Systems Development Method Agile Project Framework for Scrum") OR ("Document Title":"Scrum-of-Scrums" OR "Abstract":"Scrum-of-Scrums") OR ("Document Title":"Enterprise Scrum" OR "Abstract":"Enterprise Scrum") OR ("Document Title":"Agile Software Solution Framework" OR "Abstract":"Agile Software Solution Framework") OR ("Document Title":"Large Scale Scrum" OR "Abstract":"Large Scale Scrum") OR ("Document Title":"Scaled Agile Framework" OR "Abstract":"Scaled Agile Framework") OR ("Document Title":"Disciplined Agile 2.0" OR "Abstract":"Disciplined Agile 2.0") OR ("Document Title":"Spotify Model" OR "Abstract":"Spotify Model") OR ("Document Title":"Mega Framework" OR "Abstract":"Mega Framework") OR ("Document Title":"Enterprise Agile Delivery and Agile Governance Practice",OR "Abstract":"Enterprise Agile Delivery and Agile Governance Practice") ("Document Title":"Recipes for Agile Governance in the Enterprise",OR "Abstract":"Recipes for Agile Governance in the Enterprise",) OR ("Document Title":"Continuous Agile Framework" OR "Abstract":"Continuous Agile Framework") OR ("Document Title":"Scrum at Scale" OR "Abstract":"Scrum at Scale") OR ("Document Title":"Enterprise Transition Framework" OR "Abstract":"Enterprise Transition Framework") OR ("Document Title":"ScALeD Agile Lean Development",OR "Abstract":"ScALeD Agile Lean Development") OR ("Document Title":"eXponential Simple Continuous Autonomous Learning Ecosystem",OR "eXponential Simple Continuous Autonomous Learning Ecosystem") OR ("Document Title":"Lean Enterprise Agile Framework",OR "Abstract":"Lean Enterprise Agile Framework") OR ("Document Title":"Nexus",OR "Abstract":"Nexus") OR ("Document Title":"FAST Agile",OR "Abstract":"FAST Agile"))

Table A.3 continued from previous page

Search engine	Search string
DB2	((acmdlTitle:(+large +scale) OR recordAbstract:(+large +scale) OR acmdlTitle:(+scaling) OR recordAbstract:(+scaling)) AND (acmdlTitle:(agile agility "extreme programming" XP "feature driven development" FDD scrum crystal "pair programming" "test-driven development" TDD leanness "lean software development" "lean development" LSD) OR recordAbstract:(agile agility "extreme programming" XP "feature driven development" FDD scrum crystal "pair programming" "test-driven development" TDD leanness "lean software development" "lean development" LSD)) AND (acmdlTitle:(-manufacturing) OR recordAbstract:(-manufacturing))) OR (acmdlTitle:("Crystal Family" "Dynamic Systems Development Method Agile Project Framework for Scrum" "Scrum-of-Scrums" "Enterprise Scrum" "Agile Software Solution Framework" "Large Scale Scrum" "Scaled Agile Framework" "Disciplined Agile 2.0" "Spotify Model" "Mega Framework" "Enterprise Agile Delivery and Agile Governance Practice" "Recipes for Agile Governance in the Enterprise" "Continuous Agile Framework" "Scrum at Scale" "Enterprise Transition Framework" "ScALeD Agile Lean Development" "eXponential Simple Continuous Autonomous Learning Ecosystem" "Lean Enterprise Agile Framework" "Nexus" "FAST Agile") OR recordAbstract:("Crystal Family" "Dynamic Systems Development Method Agile Project Framework for Scrum" "Scrum-of-Scrums" "Enterprise Scrum" "Agile Software Solution Framework" "Large Scale Scrum" "Scaled Agile Framework" "Disciplined Agile 2.0" "Spotify Model" "Mega Framework" "Enterprise Agile Delivery and Agile Governance Practice" "Recipes for Agile Governance in the Enterprise" "Continuous Agile Framework" "Scrum at Scale" "Enterprise Transition Framework" "ScALeD Agile Lean Development" "eXponential Simple Continuous Autonomous Learning Ecosystem" "Lean Enterprise Agile Framework" "Nexus" "FAST Agile"))
DB4	(TS=("large scale" OR "scaling") AND (TS=("agile" OR "agility" OR "extreme programming" OR "feature driven development" OR "FDD" OR "scrum" OR "crystal" OR "pair programming" OR "test-driven development" OR "TDD" OR "leanness" OR "lean software development" OR "lean development") NOT TS=(manufacturing))) OR TS=("Crystal Family" OR "Dynamic Systems Development Method Agile Project Framework for Scrum" OR "Scrum-of-Scrums" OR "Enterprise Scrum" OR "Agile Software Solution Framework" OR "Large Scale Scrum" OR "Scaled Agile Framework" OR "Disciplined Agile 2.0" OR "Spotify Model" OR "Mega Framework" OR "Enterprise Agile Delivery and Agile Governance Practice" OR "Recipes for Agile Governance in the Enterprise" OR "Continuous Agile Framework" OR "Scrum at Scale",OR "Enterprise Transition Framework" OR "ScALeD Agile Lean Development" OR "eXponential Simple Continuous Autonomous Learning Ecosystem" OR "Lean Enterprise Agile Framework" OR "Nexus" OR "FAST Agile")

Table A.3 continued from previous page

Search engine	Search string
DB5	((title:("large scale" OR "scaling") OR abstract:("large scale" OR "scaling")) AND ((title:("agile") OR abstract:("agile")) OR (title:("agility") OR abstract:("agility")) OR (title:("extreme programming") OR abstract:("extreme programming")) OR (title:("XP") OR abstract:("XP")) OR (title:("feature driven development") OR abstract:("feature driven development")) OR (title:("FDD") OR abstract:("FDD")) OR (title:("crystal") OR abstract:("crystal")) OR (title:("air programming") OR abstract:("air programming")) OR (title:("test-driven development") OR abstract:("test-driven development")) OR (title:("TDD") OR abstract:("TDD")) OR (title:("leanness") OR abstract:("leanness")) OR (title:("lean software development") OR abstract:("lean software development")) OR (title:("lean development") OR abstract:("lean development")) OR (title:("LSD") OR abstract:("LSD"))) AND NOT ((title:("manufacturing") OR abstract:("manufacturing")))) OR ((title:("Crystal Family") OR abstract:("Crystal Family")) OR (title:("Dynamic Systems Development Method Agile Project Framework for Scrum") OR abstract:("Dynamic Systems Development Method Agile Project Framework for Scrum")) OR (title:("Scrum-of-Scrums") OR abstract:("Scrum-of-Scrums")) OR (title:("Enterprise Scrum") OR abstract:("Enterprise Scrum")) OR (title:("Agile Software Solution Framework") OR abstract:("Agile Software Solution Framework")) OR (title:("Large Scale Scrum") OR abstract:("Large Scale Scrum")) OR (title:("Scaled Agile Framework") OR abstract:("Scaled Agile Framework")) OR (title:("Disciplined Agile 2.0") OR abstract:("Disciplined Agile 2.0")) OR (title:("Spotify Model") OR abstract:("Spotify Model")) OR (title:("Mega Framework") OR abstract:("Mega Framework")) OR (title:("Enterprise Agile Delivery and Agile Governance Practice") OR abstract:("Enterprise Agile Delivery and Agile Governance Practice")) OR (title:("Recipes for Agile Governance in the Enterprise") OR abstract:("Recipes for Agile Governance in the Enterprise")) OR (title:("Continuous Agile Framework") OR abstract:("Continuous Agile Framework")) OR (title:("Scrum at Scale") OR abstract:("Scrum at Scale")) OR (title:("Enterprise Transition Framework") OR abstract:("Enterprise Transition Framework")) OR (title:("ScALeD Agile Lean Development") OR abstract:("ScALeD Agile Lean Development")) OR (title:("eXponential Simple Continuous Autonomous Learning Ecosystem") OR abstract:("eXponential Simple Continuous Autonomous Learning Ecosystem")) OR (title:("Lean Enterprise Agile Framework") OR abstract:("Lean Enterprise Agile Framework")) OR (title:("Nexus") OR abstract:("Nexus")) OR (title:("FAST Agile") OR abstract:("FAST Agile")))

Table A.3 continued from previous page

Search engine	Search string
DB6	((TITLE-ABS("large scale") OR TITLE-ABS("scaling")) AND (TITLE-ABS("agile") OR TITLE-ABS("agility") OR TITLE-ABS("extreme programming") OR TITLE-ABS("XP") OR TITLE-ABS("feature driven development") OR TITLE-ABS("FDD") OR TITLE-ABS("scrum") OR TITLE-ABS("crystal") OR TITLE-ABS("pair programming") OR TITLE-ABS("test-driven development") OR TITLE-ABS("TDD") OR TITLE-ABS("leanness") OR TITLE-ABS("lean software development") OR TITLE-ABS("lean development") OR TITLE-ABS("LSD")) AND NOT (TITLE-ABS("manufacturing"))) OR (TITLE-ABS("Crystal Family") OR TITLE-ABS("Dynamic Systems Development Method Agile Project Framework for Scrum") OR TITLE-ABS("Scrum-of-Scrums") OR TITLE-ABS("Enterprise Scrum") OR TITLE-ABS("Agile Software Solution Framework") OR TITLE-ABS("Large Scale Scrum") OR TITLE-ABS("Scaled Agile Framework") OR TITLE-ABS("Disciplined Agile 2.0") OR TITLE-ABS("Spotify Model") OR TITLE-ABS("Mega Framework") OR TITLE-ABS("Enterprise Agile Delivery and Agile Governance Practice") OR TITLE-ABS("Recipes for Agile Governance in the Enterprise") OR TITLE-ABS("Continuous Agile Framework") OR TITLE-ABS("Scrum at Scale") OR TITLE-ABS("Enterprise Transition Framework") OR TITLE-ABS("ScALeD Agile Lean Development") OR TITLE-ABS("eXponential Simple Continuous Autonomous Learning Ecosystem") OR TITLE-ABS("Lean Enterprise Agile Framework") OR TITLE-ABS("Nexus") OR TITLE-ABS("FAST Agile"))

Table A.3.: Overview of search engines and the related search terms

Bibliography

[1] W. J. Orlikowski. "Improvising Organizational Transformation Over Time: A Situated Change Perspective." In: *Information Systems Research* 7.1 (1996), pp. 63–92. DOI: 10.1287/isre.7.1.63.

[2] J. Highsmith and A. Cockburn. "Agile Software Development: The Business of Innovation." In: *IEEE Computer* 34.9 (2001), pp. 120–122. DOI: 10.1109/2.947100.

[3] P. Kettunen. "Extending Software Project Agility with New Product Development Enterprise Agility." In: *Softw. Process* 12.6 (Nov. 2007), pp. 541–548. ISSN: 1077-4866. DOI: 10.1002/spip.v12:6.

[4] T. Dingsøyr and N. B. Moe. "Towards Principles of Large-Scale Agile Development - A Summary of the Workshop at XP2014 and a Revised Research Agenda." In: *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation - XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers*. Springer. 2014, pp. 1–8. DOI: 10.1007/978-3-319-14358-3_1.

[5] M. Alqudah and R. Razali. "A review of scaling agile methods in large software development." In: *International Journal on Advanced Science, Engineering and Information Technology* 6.6 (2016), pp. 828–837. DOI: 10.18517/ijaseit.6.6.1374.

[6] T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, Ç. Gencel, and K. Petersen, eds. *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation - XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers*. Vol. 199. Lecture Notes in Business Information Processing. Springer. Springer, 2014. ISBN: 978-3-319-14357-6. DOI: 10.1007/978-3-319-14358-3.

[7] VersionOne. *10th annual state of Agile development survey*. http://www.agile247.pl/wp-content/uploads/2016/04/VersionOne-10th-Annual-State-of-Agile-Report.pdf. 2016.

[8] T. Dingsøyr and N. B. Moe. "Research challenges in large-scale agile software development." In: *ACM SIGSOFT Software Engineering Notes* 38.5 (2013), pp. 38–39. DOI: 10.1145/2507288.2507322.

[9] N. B. Moe and T. Dingsøyr. "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, May 22 - 26, 2017*. ACM. 2017, 14:1–14:4. DOI: 10.1145/3120459.3120474.

[10] A. Scheerer, T. Hildenbrand, and T. Kude. "Coordination in Large-Scale Agile Software Development: A Multiteam Systems Perspective." In: *47th Hawaii International Conference on System Sciences, HICSS 2014, Waikoloa, HI, USA, January 6-9, 2014*. IEEE. 2014, pp. 4780–4788. DOI: 10.1109/HICSS.2014.587.

[11] P. Xu. "Coordination in large agile projects." In: *Review of Business Information Systems (RBIS)* 13.4 (2009). DOI: 10.19030/rbis.v13i4.4310.

[12] M. Paasivaara and C. Lassenius. "Communities of practice in a large distributed agile software development organization – Case Ericsson." In: *Information and Software Technology* 56.12 (2014). Special issue: Human Factors in Software Development, pp. 1556–1577. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2014.06.008.

[13] D. Smite, N. B. Moe, A. Sablis, and C. Wohlin. "Software teams and their knowledge networks in large-scale software development." In: *Information & Software Technology* 86 (2017), pp. 71–86. DOI: 10.1016/j.infsof.2017.01.003.

[14] C. J. Stettina and J. Hörz. "Agile portfolio management: An empirical perspective on the practice in use." In: *International Journal of Project Management* 33.1 (2015), pp. 140–152. ISSN: 0263-7863. DOI: 10.1016/j.ijproman.2014.03.008.

[15] K. Dikert, M. Paasivaara, and C. Lassenius. "Challenges and success factors for large-scale agile transformations: A systematic literature review." In: *Journal of Systems and Software* 119 (2016), pp. 87–108. DOI: 10.1016/j.jss.2016.06.013.

[16] D. Batra, W. Xia, D. E. VanderMeer, and K. Dutta. "Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry." In: *Communications of the Association for Information Systems* 27 (2010), p. 21. DOI: 10.17705/1CAIS.02721.

[17] K. Petersen and C. Wohlin. "The effect of moving from a plan-driven to an incremental software development approach with agile practices: An industrial case study." In: *Empirical Software Engineering* 15 (Dec. 2010), pp. 654–693. DOI: 10.1007/s10664-010-9136-6.

[18] J. Vlietland and H. van Vliet. "Towards a governance framework for chains of Scrum teams." In: *Information and Software Technology* 57 (2015), pp. 52–65. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2014.08.008.

[19] T. Dingsøyr, N. B. Moe, T. E. Fægri, and E. A. Seim. "Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation." In: *Empirical Software Engineering* 23.1 (2018), pp. 490–520. DOI: 10.1007/s10664-017-9524-2.

[20] K. H. Rolland, B. Fitzgerald, T. Dingsøyr, and K. Stol. "Problematizing Agile in the Large: Alternative Assumptions for Large-Scale Agile Development." In: *Proceedings of the International Conference on Information Systems - Digital Innovation at the Crossroads, ICIS 2016, Dublin, Ireland, December 11-14, 2016*. Association for Information Systems (AIS). 2016. DOI: 10.13140/RG.2.2.27795.07207.

[21] Ö. Uludağ, M. Kleehaus, X. Xu, and F. Matthes. "Investigating the Role of Architects in Scaling Agile Frameworks." In: *21st IEEE International Enterprise Distributed Object Computing Conference, EDOC 2017, Quebec City, QC, Canada, October 10-13, 2017*. 2017, pp. 123–132. DOI: 10.1109/EDOC.2017.25.

[22] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. "Systematic Mapping Studies in Software Engineering." In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. EASE'08. Italy: BCS Learning & Development Ltd., 2008, pp. 68–77. DOI: 10.1016/j.infsof.2015.03.007.

[23] B. Kitchenham. "Procedures for performing systematic reviews." In: *Keele University Technical Report* 33.2004 (2004), pp. 1–26.

[24] B. Kitchenham, D. Budgen, and O. P. Brereton. "The value of mapping studies-A participant-observer case study." In: *14th international conference on Evaluation and Assessment in Software Engineering*. Vol. 10. 2010, pp. 25–33.

[25] P. Rodrıguez, A. Haghighatkhah, L. E. Lwakatare, S. Teppola, T. Suomalainen, J. Eskeli, T. Karvonen, P. Kuvaja, J. M. Verner, and M. Oivo. "Continuous deployment of software intensive products and services: A systematic mapping study." In: *Journal of Systems and Software* 123 (2017), pp. 263–291. DOI: 10.1016/j.jss.2015.12.015.

[26] A. Haghighatkhah, A. Banijamali, O.-P. Pakanen, M. Oivo, and P. Kuvaja. "Automotive software engineering: A systematic mapping study." In: *Journal of Systems and Software* 128 (2017), pp. 25–55. DOI: 10.1016/j.jss.2017.03.005.

[27] H. Zhang, M. A. Babar, and P. Tell. "Identifying relevant studies in software engineering." In: *Information and Software Technology* 53.6 (2011), pp. 625–637. DOI: 10.1016/j.infsof.2010.12.010.

[28] P. Runeson and M. Höst. "Guidelines for conducting and reporting case study research in software engineering." In: *Empirical software engineering* 14.2 (2009), p. 131.

[29] A. Cockburn. *Agile Software Development: The Cooperative Game*. Agile Software Development Series. Pearson Education, 2006. ISBN: 9780321630070.

[30] S. Goldman, R. Nagel, and K. Preiss. *Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer*. Industrial Engineering Series. Van Nostrand Reinhold, 1995. ISBN: 9780442019037.

[31] A. Stellman and J. Greene. *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. O'Reilly, 2014. ISBN: 9781449331924.

[32] S. Collyer, C. Warren, B. Hemsley, and C. Stevens. "Aim, Fire, Aim – Project Planning Styles in Dynamic Environments." In: *Project Management Journal* 41 (Sept. 2010), pp. 108–121. DOI: 10.1002/pmj.20199.

[33] V. Basili, B. Boehm, P. Costa, K. Dangle, F. Shull, R. Tvedt, L. Williams, and M. Zelkowitz. "Empirical Findings in Agile Methods." In: vol. 2418. Springer. Aug. 2002, pp. 197–207. DOI: 10.1007/3-540-45672-4_19.

[34] P. Serrador and J. Pinto. "Does Agile work? — A quantitative analysis of agile project success." In: *International Journal of Project Management* 33 (Mar. 2015), pp. 1040–1051. DOI: 10.1016/j.ijproman.2015.01.006.

[35] M. Coram and S. Bohner. "The impact of Agile Methods on software project management." In: May 2005, pp. 363–370. ISBN: 0-7695-2308-0. DOI: 10.1109/ECBS.2005.68.

[36] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, et al. *Manifesto for Agile Software Development*. http://agilemanifesto.org. 2001.

[37] H. Takeuchi and I. Nonaka. "The new new product development game." In: *Harvard business review* 64.1 (1986), pp. 137–146.

[38] J. Sutherland and K. Schwaber. *The definitive guide to scrum: The rules of the game.* http://www.scrum.org/scrum-guides. 2013 (Accessed on 10/16/2019).

[39] K. Schwaber. "Scrum development process." In: *Business object design and implementation*. Springer, 1997, pp. 117–134.

[40] H. F. Cervone. "Understanding agile project management methods using Scrum." In: *OCLC Systems Services* 27 (Feb. 2011), pp. 18–22. DOI: 10.1108/10650751111106528.

[41] K. Schwaber. *Agile Project Management with Scrum*. Developer Best Practices. Pearson Education, 2004. ISBN: 9780735637900.

[42] C. Keith. *Agile Game Development with Scrum*. Addison-Wesley Signature Series (Cohn). Pearson Education, 2010. ISBN: 9780321670281.

[43] K. Rubin. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley signature series. Addison-Wesley, 2012. ISBN: 9780137043293.

[44] R. Sindhgatta, N. C. Narendra, and B. Sengupta. "Software evolution in agile development: a case study." In: *Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion*. ACM. 2010, pp. 105–114. DOI: 10.1145/1869542.1869560.

[45] D. Anderson. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010. ISBN: 9780984521401.

[46] M. O. Ahmad, J. Markkula, and M. Oivo. "Kanban in software development: A systematic literature review." In: *2013 39th Euromicro conference on software engineering and advanced applications*. IEEE. 2013, pp. 9–16. DOI: 10.1109/SEAA.2013.28.

[47] M. Ikonen, P. Kettunen, N. Oza, and P. Abrahamsson. "Exploring the sources of waste in kanban software development projects." In: *36th EUROMICRO Conference on Software Engineering and Advanced Applications*. IEEE. 2010, pp. 376–381. DOI: 10.1109/SEAA.2010.40.

[48] V. Mahnic. "Improving software development through combination of scrum and kanban." In: *Recent Advances in Computer Engineering, Communications and Information Technology, Espanha* (2014), pp. 281–288.

[49] M. Yilmaz and R. V. O'Connor. "A Scrumban integrated gamification approach to guide software process improvement: a Turkish case study." In: *Tehnički vjesnik* 23.1 (2016), pp. 237–245. DOI: 10.17559/TV-20140922220409.

[50] C. Ladas. *Scrumban - Essays on Kanban Systems for Lean Software Development*. Modus Cooperandi lean series. Modus Cooperandi Press, 2009. ISBN: 9780578002149.

[51] N. Nikitina, M. Kajko-Mattsson, and M. Stråle. "From scrum to scrumban: A case study of a process transition." In: *2012 International Conference on Software and System Process (ICSSP)*. IEEE. 2012, pp. 140–149. DOI: DOI:10.1109/ICSSP.2012.6225959.

[52] A. Reddy. *The Scrumban [R]Evolution: Getting the Most Out of Agile, Scrum, and Lean Kanban*. Agile Software Development Series. Pearson Education, 2015. ISBN: 9780134077628.

[53] M. Stoica, B. Ghilic-Micu, M. Mircea, and C. Uscatu. "Analyzing Agile Development from Waterfall Style to Scrumban." In: *Informatica Economica* 20.4 (2016). DOI: 10.12948/issn14531305/20.4.2016.01.

[54] E. Corona and F. E. Pani. "A review of lean-Kanban approaches in the software development." In: *WSEAS transactions on information science and applications* 10.1 (2013), pp. 1–13.

[55] M. Poppendieck and M. A. Cusumano. "Lean software development: A tutorial." In: *IEEE software* 29.5 (2012), pp. 26–32. DOI: 10.1109/MS.2012.107.

[56] T. Dingsøyr, T. E. Fægri, and J. Itkonen. "What Is Large in Large-Scale? A Taxonomy of Scale for Agile Software Development." In: *Product-Focused Software Process Improvement - 15th International Conference, PROFES 2014, Helsinki, Finland, December 10-12, 2014. Proceedings*. Springer. 2014, pp. 273–276. DOI: 10.1007/978-3-319-13835-0_20.

[57] S. Ambler and M. Lines. "The disciplined agile process decision framework." In: *International Conference on Software Quality*. Springer. 2016, pp. 3–14. DOI: 10.1007/978-3-319-27033-3_1.

[58] S. Ambler and M. Lines. "Scaling agile software development tactically: Disciplined agile delivery at scale." In: *Disciplined Agile Consortium, Tech. Rep.* (2016).

[59] S. Ambler and M. Lines. *Scaling Factors – Disciplined Agile (DA)*. http://disciplinedagiledelivery.com/agility-at-scale/scaling-factors/. 2016 (Accessed on 08/16/2019).

[60] P. Kruchten. "Contextualizing agile software development." In: *Journal of Software: Evolution and Process* 25.4 (2013), pp. 351–361. DOI: 10.1002/smr.572.

[61] P. Kruchten. "Scaling down large projects to meet the agile "sweet spot"." In: *IBM developerWorks* 13 (2004).

[62] H. Khalid, M. Ahmed, A. Sameer, F. Arif, et al. "Systematic literature review of agile scalability for large scale projects." In: *International Journal of Advanced Computer Science and Applications (IJACSA)* 6.9 (2015), pp. 63–75. DOI: 10.14569/IJACSA.2015.060908.

[63] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil. "Lessons from applying the systematic literature review process within the software engineering domain." In: *Journal of systems and software* 80.4 (2007), pp. 571–583. DOI: 10.1016/j.jss.2006.07.009.

[64] M. F. Abrar, M. S. Khan, S. Ali, U. Ali, M. F. Majeed, A. Ali, B. Amin, and N. Rasheed. "Motivators for Large-Scale Agile Adoption From Management Perspective: A Systematic Literature Review." In: *IEEE Access* 7 (2019), pp. 22660–22674. DOI: 10.1109/ACCESS.2019.2896212.

[65] S. Keele et al. *Guidelines for performing systematic literature reviews in software engineering*. Tech. rep. Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.

[66] W. Alsaqaf, M. Daneva, and R. Wieringa. "Quality requirements in large-scale distributed agile projects–a systematic literature review." In: *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer. 2017, pp. 219–234. DOI: 10.1007/978-3-319-54045-0_17.

[67] J. Saldana. *The Coding Manual for Qualitative Researchers*. SAGE Publications, 2012. ISBN: 9781446271421.

[68] I. F. Da Silva, P. A. da Mota Silveira Neto, P. O'Leary, E. S. De Almeida, and S. R. de Lemos Meira. "Agile software product lines: a systematic mapping study." In: *Software: Practice and Experience* 41.8 (2011), pp. 899–920. DOI: 10.1002/spe.1078.

[69] D. A. Magües, J. W. Castro, and S. T. Acuña. "Usability in agile development: A systematic mapping study." In: *2016 XLII Latin American Computing Conference (CLEI)*. IEEE. 2016, pp. 1–8. DOI: 10.1109/CLEI.2016.7833347.

[70] M. Arias, M. R. Marques, and E. Rojas. "Using process mining in agile software development methodologies: a systematic mapping study." In: *2018 XLIV Latin American Computer Conference (CLEI)*. IEEE. 2018, pp. 552–561. DOI: 10.1007/978-3-030-00623-5_20.

[71] B. Kitchenham. *Guidelines for performing Systematic Literature Reviews in Software Engineering Version 2.3*. Tech. rep. Keele University and University of Durham, EBSE, 2007.

[72] K. Curcio, T. Navarro, A. Malucelli, and S. Reinehr. "Requirements engineering: A systematic mapping study in agile software development." In: *Journal of Systems and Software* 139 (2018), pp. 32–50. DOI: 10.1016/j.jss.2018.01.036.

[73] *The Chaos Report*. Tech. rep. Available at: https://www.projectsmart.co.uk/white-papers/chaos-report.pdf. Standish Group, 1995.

[74] T. Dybå and T. Dingsøyr. "Empirical studies of agile software development: A systematic review." In: *Information & Software Technology* 50.9-10 (2008), pp. 833–859. DOI: 10.1016/j.infsof.2008.01.006.

[75] C. Yang, P. Liang, and P. Avgeriou. "A systematic mapping study on the combination of software architecture and agile development." In: *Journal of Systems and Software* 111 (2016), pp. 157–184. DOI: 10.1016/j.jss.2015.09.028.

[76] S. Freudenberg and H. Sharp. "The top 10 burning research questions from practitioners." In: *IEEE Software* 27.5 (2010), pp. 8–9. DOI: 10.1109/MS.2010.129.

[77] S. L. Ram, H. Oktaba, et al. "Productivity in Agile Software Development: A Systematic Mapping Study." In: *5th International Conference in Software Engineering Research and Innovation (CONISOFT)*. IEEE. 2017, pp. 44–53. DOI: 10.1109/CONISOFT.2017.00013.

[78] B. A. Kitchenham, D. Budgen, and P. Brereton. *Evidence-based software engineering and systematic reviews*. Vol. 4. CRC press, 2015. DOI: 10.1007/11767718_3.

[79] L. R. Saldanha and A. Zorzo. *Security requirements in agille software development: A systematic mapping study*. Tech. rep. Pontifical Catholic University of Rio Grande Do Sul, 2019, 32p, 2019.

[80] K. Petersen, S. Vakkalanka, and L. Kuzniarz. "Guidelines for conducting systematic mapping studies in software engineering: An update." In: *Information and Software Technology* 64 (2015), pp. 1–18. DOI: 10.1007/s10664-008-9102-8.

[81] M. Petticrew and H. Roberts. *Systematic Reviews in the Social Sciences: A Practical Guide*. Wiley, 2008. ISBN: 9781405150149.

[82] M. A. Ali. "Survey on the state of agile practices implementation in Pakistan." In: *International Journal of Information and Communication Technology Research* 2.4 (2012).

[83] P. Rodrıguez, J. Markkula, M. Oivo, and K. Turula. "Survey on agile and lean usage in finnish software industry." In: *Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. IEEE. 2012, pp. 139–148. DOI: 10.1145/2372251.2372275.

[84] S. Stavru. "A critical examination of recent industrial surveys on agile method usage." In: *Journal of Systems and Software* 94 (2014), pp. 87–97. DOI: 10.1016/j.jss.2014.03.041.

[85] J. Stapleton. *DSDM, Dynamic Systems Development Method: The Method in Practice*. Addison-Wesley, 1997. ISBN: 9780201178890.

[86] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. "Requirements engineering paper classification and evaluation criteria: A proposal and a discussion." In: *Requirements engineering* 11.1 (2006), pp. 102–107. DOI: 10.1007/s00766-005-0021-6.

[87] V. Berg, J. Birkeland, A. Nguyen-Duc, I. O. Pappas, and L. Jaccheri. "Software startup engineering: A systematic mapping study." In: *Journal of Systems and Software* 144 (2018), pp. 255–274. DOI: 10.1016/j.jss.2018.06.043.

[88] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson. "Software development in startup companies: A systematic mapping study." In: *Information and Software Technology* 56.10 (2014), pp. 1200–1218. DOI: 10.1016/j.infsof.2014.04.014.

[89] M. Ivarsson and T. Gorschek. "A method for evaluating rigor and industrial relevance of technology evaluations." In: *Empirical Software Engineering* 16.3 (2011), pp. 365–395. DOI: 10.1007/s10664-010-9146-4.

[90] Z. Li, P. Avgeriou, and P. Liang. "A systematic mapping study on technical debt and its management." In: *Journal of Systems and Software* 101 (2015), pp. 193–220. DOI: 10.1016/j.jss.2014.12.027.

[91] K. Petersen and C. Wohlin. "A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case." In: *Journal of systems and software* 82.9 (2009), pp. 1479–1490. DOI: 10.1016/j.jss.2009.03.036.

[92] M. Laanti, O. Salo, and P. Abrahamsson. "Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation." In: *Information & Software Technology* 53.3 (2011), pp. 276–290. DOI: 10.1016/j.infsof.2010.11.010.

[93] D. Gerster, C. Dremel, and P. Kelker. ""Agile meets non-agile": Implications of adopting agile practices at enterprises." In: *24th Americas Conference on Information Systems*. 2018.

[94] 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*. ACM. 2011, p. 3. DOI: 10.1145/2068783.2068786.

[95] A. Qumer and B. Henderson-Sellers. "A framework to support the evaluation, adoption and improvement of agile methods in practice." In: *Journal of Systems and Software* 81.11 (2008), pp. 1899–1919. DOI: 10.1016/j.jss.2007.12.806.

[96] I. Stojanov, O. Turetken, and J. J. Trienekens. "A maturity model for scaling agile development." In: *41st Euromicro Conference on Software Engineering and Advanced Applications*. IEEE. 2015, pp. 446–453. DOI: 10.1109/SEAA.2015.29.

[97] K. Power. "A model for understanding when scaling agile is appropriate in large organizations." In: *International Conference on Agile Software Development*. Springer. 2014, pp. 83–92. DOI: 10.1007/978-3-319-14358-3_8.

[98] V. Heikkilä, K. Rautiainen, and S. Jansen. "A Revelatory Case Study on Scaling Agile Release Planning." In: *36th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2010, Lille, France, September 1-3, 2010*. IEEE. 2010, pp. 289–296. DOI: 10.1109/SEAA.2010.37.

[99] M. Shameem, B. Chandra, R. R. Kumar, and C. Kumar. "A systematic literature review to identify human related challenges in globally distributed agile software development: towards a hypothetical model for scaling agile methodologies." In: *2018 4th International Conference on Computing Communication and Automation (ICCCA)*. IEEE. 2018, pp. 1–7. DOI: 10.1109/CCAA.2018.8777533.

[100] R. Duijs, P. Ravesteyn, and M. van Steenbergen. "Adaptation of enterprise architecture efforts to an agile environment." In: *Bled eConference*. 2018, p. 16. DOI: 10.18690/978-961-286-170-4.26.

[101] R. Vallon, C. Dräger, 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*. IEEE. 2014, pp. 52–60. DOI: 10.1109/AGILE.2014.22.

[102] M. Paasivaara. "Adopting SAFe to Scale Agile in a Globally Distributed Organization." In: *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*. IEEE. May 2017, pp. 36–40. DOI: 10.1109/ICGSE.2017.15.

[103] J. M. Bass. "Agile Method Tailoring in Distributed Enterprises: Product Owner Teams." In: *8th IEEE International Conference on Global Software Engineering, ICGSE 2013, Bari, Italy, August 26-29, 2013*. IEEE. 2013, pp. 154–163. DOI: 10.1109/ICGSE.2013.27.

[104] B. Horlach, I. Schirmer, and P. Drews. "Agile Portfolio Management: Design Goal and Principles." In: *Proceedings of the European Conference on Information Systems, Stockholm-Uppsala*. AIS Electronic Library (AISeL). 2019.

[105] A. Šāblis and D. Šmite. "Agile Teams in Large-Scale Distributed Context: Isolated or Connected?" In: *Proceedings of the Scientific Workshop Proceedings of XP2016*. ACM. 2016, p. 10. DOI: 10.1145/2962695.2962705.

[106] S. Theobald, A. Schmitt, and P. Diebold. "Comparing scaling agile frameworks based on underlying practices." In: *International Conference on Agile Software Development*. Springer. 2019, pp. 88–96. DOI: 10.1007/978-3-030-30126-2_11.

[107] J. M. Bass. "Artefacts and agile method tailoring in large-scale offshore software development programmes." In: *Information & Software Technology* 75 (2016), pp. 1–16. DOI: 10.1016/j.infsof.2016.03.001.

[108] O. Turetken, I. Stojanov, and J. J. Trienekens. "Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework." In: *Journal of Software: Evolution and Process* 29.6 (2017), e1796. DOI: 10.1002/smr.1796.

[109] T. Gustavsson. "Assigned roles for Inter-team coordination in Large-Scale Agile Development: A literature review." In: *Proceedings of the XP2017 Scientific Workshops*. ACM. 2017, p. 15. DOI: 10.1145/3120459.3120475.

[110] C. Fuchs and T. Hess. "Becoming agile in the digital transformation: The process of a large-scale agile transformation." In: *39th International Conference On Information Systems, San Francisco*. 2018.

[111] P. Kettunen and M. Laanti. "Combining agile software projects and large-scale organizational agility." In: *Software Process: Improvement and Practice* 13.2 (2008), pp. 183–193. DOI: 10.1002/spip.354.

[112] A. Martini, L. Pareto, and J. Bosch. "Communication factors for speed and reuse in large-scale agile software development." In: *Proceedings of the 17th international software product line conference*. ACM. 2013, pp. 42–51. DOI: 10.1145/2491627.2491642.

[113] D. Mishra and A. Mishra. "Complex software project development: agile methods adoption." In: *Journal of Software Maintenance and Evolution: Research and Practice* 23.8 (2011), pp. 549–564. DOI: 10.1002/smr.528.

[114] V. Heikkilä, M. Paasivaara, C. Lassenius, and C. Engblom. "Continuous Release Planning in a Large-Scale Scrum Development Organization at Ericsson." In: *Agile Processes in Software Engineering and Extreme Programming - 14th International Conference, XP 2013, Vienna, Austria, June 3-7, 2013. Proceedings*. Springer. 2013, pp. 195–209. DOI: 10.1007/978-3-642-38314-4_14.

[115] T. Dingsøyr, N. B. Moe, and E. A. Seim. "Coordinating Knowledge Work in Multi-Team Programs: Findings from a Large-Scale Agile Development Program." In: *Project Management Journal* 49.6 (2018). arXiv: 1801.08764.

[116] S. Bick, K. Spohrer, R. Hoda, A. Scheerer, and A. Heinzl. "Coordination Challenges in Large-Scale Software Development: A Case Study of Planning Misalignment in Hybrid Settings." In: *IEEE Trans. Software Eng.* 44.10 (2018), pp. 932–950. DOI: 10.1109/TSE.2017.2730870.

[117] 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." In: *Procedia Computer Science* 121 (2017), pp. 123–128. DOI: 10.1016/j.procs.2017.11.017.

[118] S. K. Weiss and P. Brune. "Crossing the Boundaries - Agile Methods in Large-Scale, Plan-Driven Organizations: A Case Study from the Financial Services Industry." In: *Advanced Information Systems Engineering - 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings*. Springer. 2017, pp. 380–393. DOI: 10.1007/978-3-319-59536-8_24.

[119] H. H. Olsson, J. Bosch, and H. Alahyari. "Customer-specific teams for agile evolution of large-scale embedded systems." In: *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*. IEEE. 2013, pp. 82–89. DOI: 10.1109/SEAA.2013.43.

[120] M. Paasivaara, S. Durasiewicz, and C. Lassenius. "Distributed Agile Development: Using Scrum in a Large Project." In: *3rd IEEE International Conference on Global Software Engineering, ICGSE 2008, Bangalore, India, 17-20 August, 2008*. IEEE. 2008, pp. 87–95. DOI: 10.1109/ICGSE.2008.38.

[121] Ö. Uludağ, N.-M. Harders, and F. Matthes. "Documenting Recurring Concerns and Patterns in Large-Scale Agile Development." In: *24th European Conference on Pattern Languages of Programs*. 2019.

[122] T. Gustavsson. "Dynamics of Inter-Team Coordination Routines in Large-Scale Agile Software Development." In: *27th European Conference on Information Systems - Information Systems for a Sharing Society, ECIS 2019, Stockholm and Uppsala, Sweden, June 8-14, 2019*. 2019. ISBN: 978-1-7336325-0-8.

[123] M. Usman, R. Britto, L.-O. Damm, and J. Börstler. "Effort estimation in large-scale software development: An industrial case study." In: *Information and Software Technology* 99 (2018), pp. 21–40. DOI: 10.1016/j.infsof.2018.02.009.

[124] M. Paasivaara and C. Lassenius. "Empower Your Agile Organization: Community-Based Decision Making in Large-Scale Agile Development at Ericsson." In: *IEEE Software* 36.2 (2019), pp. 64–69. DOI: 10.1109/MS.2018.2886827.

[125] J. H. Gundelsby. "Enabling autonomous teams in large-scale agile through architectural principles." In: *Proceedings of the 19th International Conference on Agile Software Development: Companion*. ACM. 2018, p. 17. DOI: 10.1145/3234152.3234183.

[126] R. Lal and T. Clear. "Enhancing product and service capability through scaling agility in a global software vendor environment." In: *Proceedings of the 13th Conference on Global Software Engineering*. ACM. 2018, pp. 59–68. DOI: 10.1145/3196369.3196378.

[127] Ö. Uludağ, S. Nägele, and M. Hauder. "Establishing Architecture Guidelines in Large-Scale Agile Development Through Institutional Pressures: A Single-Case Study." In: *25th Americas Conference on Information Systems, AMCIS 2019, Cancún, Mexico, August 15-17, 2019*. 2019.

[128] M. Paasivaara, V. Heikkilä, and C. Lassenius. "Experiences in Scaling the Product Owner Role in Large-Scale Globally Distributed Scrum." In: *2012 IEEE Seventh International Conference on Global Software Engineering, Porto Alegre, Rio Grande do Sul, Brazil, August 27-30, 2012*. IEEE. 2012, pp. 174–178. DOI: 10.1109/ICGSE.2012.41.

[129] V. Santos, A. Goldman, and C. R. De Souza. "Fostering effective inter-team knowledge sharing in agile software development." In: *Empirical Software Engineering* 20.4 (2015), pp. 1006–1051. DOI: 10.1007/s10664-014-9307-y.

[130] R. M. Parizi, T. J. Gandomani, and M. Z. Nafchi. "Hidden facilitators of agile transition: Agile coaches and agile champions." In: *2014 8th. Malaysian Software Engineering Conference (MySEC)*. IEEE. 2014, pp. 246–250. DOI: 10.1109/MySec.2014.6986022.

[131] D. Gerster, C. Dremel, W. Brenner, and P. Kelker. "How Enterprises Adopt Agile Structures: A Multiple-Case Study." In: *52nd Hawaii International Conference on System Sciences, HICSS 2019, Grand Wailea, Maui, Hawaii, USA, January 8-11, 2019*. 2019, pp. 1–10.

[132] Ö. Uludağ, M. Kleehaus, C. Caprano, and F. Matthes. "Identifying and structuring challenges in large-scale agile development based on a structured literature review." In: *2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC)*. IEEE. 2018, pp. 191–197. DOI: 10.1109/EDOC.2018.00032.

[133] T. Gustavsson. "Impacts on team performance in large-scale agile software development." In: *2018 Joint of the 17th Business Informatics Research Short Papers, Workshops and Doctoral Consortium, BIR-WS 2018, 24 September 2018 through 26 September 2018*. CEUR-WS. 2018, pp. 421–431.

[134] K. Conboy and N. Carroll. "Implementing Large-Scale Agile Frameworks: Challenges and Recommendations." In: *IEEE Software* 36.2 (2019), pp. 44–50.

[135] M. Paasivaara, C. Lassenius, V. Heikkilä, K. Dikert, and C. Engblom. "Integrating Global Sites into the Lean and Agile Transformation at Ericsson." In: *8th IEEE International Conference on Global Software Engineering, ICGSE 2013, Bari, Italy, August 26-29, 2013*. IEEE. 2013, pp. 134–143. DOI: 10.1109/ICGSE.2013.25.

[136] 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, Edinburgh, Scotland, UK, May 24, 2016*. ACM. 2016, p. 4. DOI: 10.1145/2962695.2962699.

[137] F. O. Bjørnson, J. Wijnmaalen, C. J. Stettina, and T. Dingsøyr. "Inter-team Coordination in Large-Scale Agile Development: A Case Study of Three Enabling Mechanisms." In: *Agile Processes in Software Engineering and Extreme Programming - 19th International Conference, XP 2018, Porto, Portugal, May 21-25, 2018, Proceedings*. Springer, Cham. 2018, pp. 216–231. DOI: 10.1007/978-3-319-91602-6_15.

[138] 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*. ACM. 2012, pp. 235–238. DOI: 10.1145/2372251.2372294.

[139] H. Nyrud and V. Stray. "Inter-team coordination mechanisms in large-scale agile." In: *Proceedings of the XP2017 Scientific Workshops*. ACM. 2017, p. 16. DOI: 10.1145/3120459.3120476.

[140] Ö. Uludağ, M. Kleehaus, N. Dreymann, C. Kabelin, and F. Matthes. "Investigating the adoption and application of large-scale scrum at a German automobile manufacturer." In: *Proceedings of the 14th International Conference on Global Software Engineering, ICGSE 2019, Montreal, QC, Canada, May 25-31, 2019*. IEEE Press. 2019, pp. 22–29. DOI: 10.1109/ICGSE.2019.00019.

[141] Ö. Uludağ, H. A. Proper, and F. Matthes. "Investigating the Establishment of Architecture Principles for Supporting Large-Scale Agile Transformations." In: *2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC)*. 2019, p. 5.

[142] P. Lous, M. Kuhrmann, and P. Tell. "Is scrum fit for global software engineering?" In: *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*. IEEE. 2017, pp. 1–10. DOI: 10.1109/ICGSE.2017.13.

[143] 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*. ACM. 2016, p. 7. DOI: 10.1145/2962695.2962702.

[144] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen. "Large-scale agile transformation at Ericsson: a case study." In: *Empirical Software Engineering* 23.5 (2018), pp. 2550–2596. DOI: 10.1007/s10664-017-9555-8.

[145] J. M. Bass. "Large-Scale Offshore Agile Tailoring: Exploring Product and Service Organisations." In: *Proceedings of the Scientific Workshop Proceedings of XP2016, Edinburgh, Scotland, UK, May 24, 2016*. ACM. 2016, p. 8. DOI: 10.1145/2962695.2962703.

[146] T. Dingsøyr, M. Mikalsen, A. Solem, and K. Vestues. "Learning in the Large - An Exploratory Study of Retrospectives in Large-Scale Agile Development." In: *Agile Processes in Software Engineering and Extreme Programming - 19th International Conference, XP 2018, Porto, Portugal, May 21-25, 2018, Proceedings*. Springer, Cham. 2018, pp. 191–198. DOI: 10.1007/978-3-319-91602-6_13.

[147] V. T. Heikkilä, M. Paasivaara, C. Lassenius, D. E. Damian, and C. Engblom. "Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson." In: *Empirical Software Engineering* 22.6 (2017), pp. 2892–2936. DOI: 10.1007/s10664-016-9491-z.

[148] K. Korhonen. "Migrating defect management from waterfall to agile software development in a large-scale multi-site organization: A case study." In: *International Conference on Agile Processes and Extreme Programming in Software Engineering*. Springer. 2009, pp. 73–82. DOI: 10.1007/978-3-642-01853-4_10.

[149] G. Papadopoulos. "Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects." In: *Procedia-Social and Behavioral Sciences* 175 (2015), pp. 455–463. DOI: 10.1016/j.sbspro.2015.01.1223.

[150] N. B. Moe, D. Smite, A. Sablis, A. Börjesson, and P. Andréasson. "Networking in a large-scale distributed agile project." In: *2014 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '14, Torino, Italy, September 18-19, 2014*. ACM. 2014, 12:1–12:8. DOI: 10.1145/2652524.2652584.

[151] V. Heikkilä, M. Paasivaara, K. Rautiainen, C. Lassenius, T. Toivola, and J. Järvinen. "Operational release planning in large-scale Scrum with multiple stakeholders - A longitudinal case study at F-Secure Corporation." In: *Information & Software Technology* 57 (2015), pp. 116–140. DOI: 10.1016/j.infsof.2014.09.005.

[152] D. Badampudi, S. A. Fricker, and A. M. Moreno. "Perspectives on Productivity and Delays in Large-Scale Agile Projects." In: *International Conference on Agile Software Development*. Springer. 2013, pp. 180–194. DOI: 10.1007/978-3-642-38314-4_13.

[153] V. Stray. "Planned and unplanned meetings in large-scale projects." In: *Proceedings of the 19th International Conference on Agile Software Development: Companion*. ACM. 2018, p. 8. DOI: 10.1145/3234152.3234178.

[154] T. Gustavsson. "Practices for Vertical and Horizontal Coordination in the Scaled Agile Framework." In: *Information Systems Development: Designing Digitalization, ISD 2018 Proceedings, Lund, Sweden August 22-24, 2018*. 2018.

[155] M. Shameem, R. R. Kumar, C. Kumar, B. Chandra, and A. A. Khan. "Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process." In: *Journal of Software: Evolution and Process* 30.11 (2018). DOI: 10.1002/smr.1979.

[156] T. E. Fægri and N. B. Moe. "Re-conceptualizing requirements engineering: findings from a large-scale, agile project." In: *Scientific Workshop Proceedings of the XP2015, Helsinki, Finland, May 25-29, 2015*. ACM. 2015, p. 4. DOI: 10.1145/2764979.2764983.

[157] M. Jorgensen. "Relationships between Project Size, Agile Practices, and Successful Software Development: Results and Analysis." In: *IEEE Software* 36.2 (2019), pp. 39–43. DOI: 10.1109/MS.2018.2884863.

[158] 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, Edinburgh, Scotland, UK, May 24, 2016*. ACM. 2016, p. 5. DOI: 10.1145/2962695.2962700.

[159] P. Diebold, A. Schmitt, and S. Theobald. "Scaling agile: how to select the most appropriate framework." In: *Proceedings of the 19th International Conference on Agile Software Development: Companion*. ACM. 2018, p. 7. DOI: 10.1145/3234152.3234177.

[160] M. Kalenda, P. Hyna, and B. Rossi. "Scaling agile in large organizations: Practices, challenges, and success factors." In: *Journal of Software: Evolution and Process* 30.10 (2018), e1954. DOI: 10.1002/smr.1954.

[161] M. Paasivaara and C. Lassenius. "Scaling Scrum in a Large Globally Distributed Organization: A Case Study." In: *11th IEEE International Conference on Global Software Engineering, ICGSE 2016, Orange County, CA, USA, August 2-5, 2016*. IEEE. 2016, pp. 74–83. DOI: 10.1109/ICGSE.2016.34.

[162] J. M. Bass. "Scrum Master Activities: Process Tailoring in Large Enterprise Projects." In: *IEEE 9th International Conference on Global Software Engineering, ICGSE 2014, Shanghai, China, 18-21 August, 2014*. IEEE. 2014, pp. 6–15. DOI: 10.1109/ICGSE.2014.24.

[163] R. Britto, D. Smite, and L. Damm. "Software Architects in Large-Scale Distributed Projects: An Ericsson Case Study." In: *IEEE Software* 33.6 (2016), pp. 48–55. DOI: 10.1109/MS.2016.146.

[164] H. Tendedez, M. A. M. Ferrario, and J. Whittle. "Software Development and CSCW: Standardization and Flexibility in Large-Scale Agile Development." In: *Proceedings of the ACM on Human-Computer Interaction* 2.CSCW (2018), p. 171. DOI: 10.1145/3274440.

[165] D. Smite, N. B. Moe, G. Levinta, and M. Floryan. "Spotify Guilds: How to Succeed With Knowledge Sharing in Large-Scale Agile Organizations." In: *IEEE Software* 36.2 (2019), pp. 51–57. DOI: 10.1109/MS.2018.2886178.

[166] B. S. Blau, T. Hildenbrand, R. Knapper, A. Mazarakis, Y. Xu, and M. G. Fassunge. "Steering through Incentives in Large-Scale Lean Software Development." In: *Evaluation of Novel Approaches to Software Engineering - 6th International Conference, ENASE 2011, Beijing, China, June 8-11, 2011. Revised Selected Papers*. Springer. 2011, pp. 32–48. DOI: 10.1007/978-3-642-32341-6_3.

[167] M. Paasivaara, O. Väättänen, M. Hallikainen, and C. Lassenius. "Supporting a Large-Scale Lean and Agile Transformation by Defining Common Values." In: *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation - XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers*. Springer. 2014, pp. 73–82. DOI: 10.1007/978-3-319-14358-3_7.

[168] Ö. Uludağ, M. Hauder, M. Kleehaus, C. Schimpfle, and F. Matthes. "Supporting Large-Scale Agile Development with Domain-Driven Design." In: *Agile Processes in Software Engineering and Extreme Programming - 19th International Conference, XP 2018, Porto, Portugal, May 21-25, 2018, Proceedings*. Springer. 2018, pp. 232–247. DOI: 10.1007/978-3-319-91602-6_16.

[169] 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*. IEEE. 2011, pp. 1–10. DOI: 10.1109/HICSS.2011.390.

[170] M. Shameem, C. Kumar, B. Chandra, and A. A. Khan. "Systematic Review of Success Factors for Scaling Agile Methods in Global Software Development Environment: A Client-Vendor Perspective." In: *24th Asia-Pacific Software Engineering Conference Workshops, APSEC Workshops 2017, Nanjing, China, December 4-8, 2017*. IEEE. 2017, pp. 17–24. DOI: 10.1109/APSECW.2017.22.

[171] J. M. Bass and A. Haxby. "Tailoring Product Ownership in Large-Scale Agile Projects: Managing Scale, Distance, and Governance." In: *IEEE Software* 36.2 (2019), pp. 58–63. DOI: 10.1109/MS.2018.2885524.

[172] N. Sekitoleko, F. Evbota, E. Knauss, A. Sandberg, M. Chaudron, and H. H. Olsson. "Technical Dependency Challenges in Large-Scale Agile Software Development." In: *International Conference on Agile Software Development*. Springer. 2014, pp. 46–61. DOI: 10.1007/978-3-319-06862-6_4.

[173] A. Scheerer, S. Bick, T. Hildenbrand, and A. Heinzl. "The Effects of Team Backlog Dependencies on Agile Multiteam Systems: A Graph Theoretical Approach." In: *48th Hawaii International Conference on System Sciences, HICSS 2015, Kauai, Hawaii, USA, January 5-8, 2015*. IEEE. 2015, pp. 5124–5132. DOI: 10.1109/HICSS.2015.606.

[174] L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl, and D. Ståhl. "The Impact of Agile Principles and Practices on Large-Scale Software Development Projects: A Multiple-Case Study of Two Projects at Ericsson." In: *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. IEEE. 2013, pp. 348–356. DOI: 10.1109/ESEM.2013.53.

[175] J. Pries-Heje and M. M. Krohn. "The SAFe Way to the Agile Organization." In: *Proceedings of the XP2017 Scientific Workshops*. ACM. 2017, p. 18. DOI: 10.1145/3120459.3120478.

[176] H. Berger and P. Beynon-Davies. "The utility of rapid application development in large-scale, complex projects." In: *Information Systems Journal* 19.6 (2009), pp. 549–570. DOI: 10.1111/j.1365-2575.2009.00329.x.

[177] N. B. Moe, T. Dingsøyr, and K. Rolland. "To schedule or not to schedule? An investigation of meetings as an inter-team coordination mechanism in large-scale agile software development." In: *IJISPM - International Journal of Information Systems and Project Management* 6.3 (2018), pp. 45–59. DOI: 10.12821/ijispm060303.

[178] H. H. Olsson and J. Bosch. "Towards continuous validation of customer value." In: *Scientific Workshop Proceedings of the XP2015*. ACM. 2015, p. 3. DOI: 10.1145/2764979.2764982.

[179] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen. "Towards Rapid Releases in Large-Scale XaaS Development at Ericsson: A Case Study." In: *IEEE 9th International Conference on Global Software Engineering, ICGSE 2014, Shanghai, China, 18-21 August, 2014*. IEEE. 2014, pp. 16–25. DOI: 10.1109/ICGSE.2014.22.

[180] M. Jovanović, A. Mas, A.-L. Mesquida, and B. Lalić. "Transition of organizational roles in Agile transformation process: A grounded theory approach." In: *Journal of Systems and Software* 133 (2017), pp. 174–194. DOI: 10.1016/j.jss.2017.07.008.

[181] W. Alsaqaf, M. Daneva, and R. Wieringa. "Understanding Challenging Situations in Agile Quality Requirements Engineering and Their Solution strategies: Insights from a Case Study." In: *2018 IEEE 26th International Requirements Engineering Conference (RE)*. IEEE. 2018, pp. 274–285. DOI: 10.1109/RE.2018.00035.

[182] B. V. Thummadi, V. D. Khapre, and R. J. Ocker. "Unpacking Agile Enterprise Architecture Innovation work practices: A Qualitative Case Study of a Railroad Company." In: *23rd Americas Conference on Information Systems, AMCIS 2017, Boston, MA, USA, August 10-12, 2017*. 2017.

[183] M. Paasivaara, S. Durasiewicz, and C. Lassenius. "Using scrum in a globally distributed project: a case study." In: *Software Process: Improvement and Practice* 13.6 (2008), pp. 527–544. DOI: 10.1002/spip.402.

[184] Ö. Uludağ, M. Kleehaus, S. Erçelik, and F. Matthes. "Using Social Network Analysis to Investigate the Collaboration Between Architects and Agile Teams: A Case Study of a Large-Scale Agile Development Program in a German Consumer Electronics Company." In: *Agile Processes in Software Engineering and Extreme Programming - 20th International Conference, XP 2019, Montréal, QC, Canada, May 21-25, 2019, Proceedings*. 2019, pp. 137–153. DOI: 10.1007/978-3-030-19034-7_9.

[185] Ö. Uludağ, M. Kleehaus, N. Reiter, and F. Matthes. "What to Expect from Enterprise Architects in Large-Scale Agile Development? A Multiple-Case Study." In: *25th Americas Conference on Information Systems, AMCIS 2019, Cancún, Mexico, August 15-17, 2019*. 2019.

[186] T. Dingsøyr and N. B. Moe. "Research challenges in large-scale agile software development." In: *ACM SIGSOFT Software Engineering Notes* 38.5 (2013), pp. 38–39. DOI: 10.1145/2507288.2507322.

[187] T. Dingsøyr and N. B. Moe. "Towards Principles of Large-Scale Agile Development - A Summary of the Workshop at XP2014 and a Revised Research Agenda." In: *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation - XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers*. 2014, pp. 1–8. DOI: 10.1007/978-3-319-14358-3_1.

[188] 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, Edinburgh, Scotland, UK, May 24, 2016*. 2016, p. 1. DOI: 10.1145/2962695.2962696.

[189] N. B. Moe and T. Dingsøyr. "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, May 22 - 26, 2017*. 2017, 14:1–14:4. DOI: 10.1145/3120459.3120474.

[190] T. Dingsøyr, N. B. Moe, and H. H. Olsson. "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, XP 2019, Companion, Porto, Portugal, May 21-25, 2018*. 2018, 6:1–6:4. DOI: 10.1145/3234152.3234176.

[191] J. M. Bass. "Future Trends in Agile at Scale: A Summary of the 7th International Workshop on Large-Scale Agile Development." In: *Agile Processes in Software Engineering and Extreme Programming - Workshops - XP 2019 Workshops, Montréal, QC, Canada, May 21-25, 2019, Proceedings*. 2019, pp. 75–80. DOI: 10.1007/978-3-030-30126-2_9.

[192] L. Barroca, T. Dingsøyr, and M. Mikalsen. "Agile Transformation: A Summary and Research Agenda from the First International Workshop." In: *Agile Processes in Software Engineering and Extreme Programming - Workshops - XP 2019 Workshops, Montréal, QC, Canada, May 21-25, 2019, Proceedings*. 2019, pp. 3–9. DOI: 10.1007/978-3-030-30126-2_1.

[193] V. Stray, N. B. Moe, and R. Hoda. "Autonomous agile teams: challenges and future directions for research." In: *Proceedings of the 19th International Conference on Agile Software Development, XP 2019, Companion, Porto, Portugal, May 21-25, 2018*. 2018, 16:1–16:5. DOI: 10.1145/3234152.3234182.

[194] N. B. Moe, V. Stray, and R. Hoda. "Trends and Updated Research Agenda for Autonomous Agile Teams: A Summary of the Second International Workshop at XP2019." In: *Agile Processes in Software Engineering and Extreme Programming - Workshops - XP 2019 Workshops, Montréal, QC, Canada, May 21-25, 2019, Proceedings*. 2019, pp. 13–19. DOI: 10.1007/978-3-030-30126-2_2.