ABSTRACT

Context: Distributed teams characterize Global Software Engineering (GSE). GSE stakeholders are from different cultures, geographic places and potentially time zones. These conditions have significant consequences on communication, coordination and control of software projects. Given these constraints, distributed teams need to highly rely on each other. Trust is the glue that holds them together and enables more open communication, which increases their performance and quality of delivered products. Simultaneously, in striving for more efficient software development approaches, Agile values and principles were formulated. Agile methods encourage establishing close collaboration between customers and developers, continuous requirements gathering and frequent face-to-face communications.

Objective: The major objective of the research is to study efficient software development approaches particularly in (globally) distributed settings. Thus, the dynamics of trust in GSE are investigated for bringing useful trust improvement suggestions to project managers. Furthermore, Agile practices that have been efficiently applied in GSE are identified through two different systematic literature review approaches (i.e. systematic literature review and backward snowballing). The differences identified in the use of Agile practices lead to a need to better understand and assess Agility.

Method: The research methods, include systematic literature reviews and case studies, are applied in different empirical cases. Then, a variety of secondary data collection methods are utilized such as semi-structured interviews, questionnaires, open discussions and presentations.

Result: Achieving trust was realized to be crucial and the success factor for trust was the “awareness” of particular GSE challenges, which shall be communicated properly to all distributed team members and proper actions shall be taken to address them. Besides, the literature indicated several successful combinations of Agile and GSE. However, despite utilizing two different literature search methods the identified patterns were similar. The most common practices were “standup meetings” and “sprints/iterations”.

Nevertheless, the current literature reports “Agile” as a general term and “distributed team” as the most common team/organization setting, which motivated examining the applicability of existing Agile assessment tools in an industrial setting. We found one of the studied tools sufficiently applicable in the context of the case organization.

Conclusions: Trust achievement is crucial for efficient GSE collaborations regardless of the applied software development approach. Although Agile promotes trust among team members, it was formulated without considering teams’ distribution. Hence, combining Agile and GSE is challenging. The literature contains several successful cases of implementing Agile in GSE while practitioners and researchers are not yet consistent regarding their perception of Agile practices and documenting them. Therefore, they need to collaborate closely, illustrate the practices, agree on the terminology, how to document the context, and how to profile/assess Agility. For this purpose, we examined the applicability of a set of Agile assessment tools and proposed one tool for the case organization.
Efficient Software Development Through Agile Methods

Samireh Jalali
Efficient Software Development Through Agile Methods

Samireh Jalali

Licentiate Dissertation in Software Engineering

School of Computing
Blekinge Institute of Technology
SWEDEN
“To improve is to change, to be perfect is to change often.”

Winston Churchill
Abstract

Context: Distributed teams characterize Global Software Engineering (GSE). GSE stakeholders are from different cultures, geographic places, and potentially time zones. These conditions have significant consequences on communication, coordination, and control of software projects. Given these constraints, distributed teams need to highly rely on each other. Trust is the glue that holds them together and enables more open communication, which increases their performance and quality of delivered products.

Simultaneously, in striving for more efficient software development approaches, Agile values and principles were formulated. Agile methods encourage establishing close collaboration between customers and developers, continuous requirements gathering, and frequent face-to-face communications.

Objective: The major objective of the research is to study efficient software development approaches particularly in (globally) distributed settings. Thus, the dynamics of trust in GSE are investigated for bringing useful trust improvement suggestions to project managers. Furthermore, Agile practices that have been efficiently applied in GSE are identified through two different systematic literature review approaches (i.e. systematic literature review and backward snowballing). The differences identified in the use of Agile practices lead to a need to better understand and assess Agility.

Method: The research methods, include systematic literature reviews and case studies, are applied in different empirical cases. Then, a variety of secondary data collection methods are utilized such as semi-structured interviews, questionnaires, open discussions, and presentations.

Result: Achieving trust was realized to be crucial and the success factor for trust was the “awareness” of particular GSE challenges, which shall be communicated properly to all distributed team members and proper actions shall be taken to address them.

Besides, the literature indicated several successful combinations of Agile and GSE. However, despite utilizing two different literature search methods the identified patterns were similar. The most common practices were “standup meetings” and “sprints/iterations”.

Nevertheless, the current literature reports “Agile” as a general term and “distributed team” as the most common team/organization setting, which motivated examining the applicability of existing Agile assessment tools in an industrial setting. We found one of the studied tools sufficiently applicable in the context of the case organization.

Conclusions: Trust achievement is crucial for efficient GSE collaborations regardless of the applied software development approach. Although Agile promotes trust among team members, it was formulated without considering teams’ distribution. Hence, combining Agile and GSE is challenging.

The literature contains several successful cases of implementing Agile in GSE while practitioners and researchers are not yet consistent regarding their perception of Agile practices and documenting them.

Therefore, they need to collaborate closely, illustrate the practices, agree on the terminology, how to document the context, and how to profile/assess Agility. For this purpose, we examined the applicability of a set of Agile assessment tools and proposed one tool for the case organization.
Acknowledgements

First, I would like to sincerely thank my supervisor, Prof. Claes Wohlin, for his invaluable feedback, expertise and advice. I appreciate that he has always been available to me to ensure I get the support I need besides his busy schedule.

Recognition must also be given to my colleagues in the SERL group for creating a positive and supportive research environment. I would like to extend special thanks to my collaborators – Dr. Cigdem Gencel, Dr. Darja Šmite, and my co-supervisor, Dr. Richard Torkar.

I am grateful to everyone who has participated in this research – filling in questionnaires, providing feedback, and putting me in contact with the right people. Special thanks must be extended to Softhouse, and in particular my contact persons, for continued support of my research.

Finally I would like to thank my family for their continued support and love throughout my life.

This work was funded by the Industrial Excellence Center EASE - Embedded Applications Software Engineering. (http://ease.cs.lth.se).
Papers in this Thesis


Related Papers not in this Thesis

Table of Contents

Chapter 1 ....................................................................................................................... 1
1.1. Overview ............................................................................................................... 1
1.2. Background .......................................................................................................... 2
  1.2.1. Global Software Engineering ................................................................. 2
  1.2.2. Agile Software Development ................................................................. 3
1.3. Research Gaps and Contributions ..................................................................... 4
1.4. Research Questions ............................................................................................. 7
1.5. Research Methodology ....................................................................................... 10
  1.5.1. Research Design ....................................................................................... 10
  1.5.2. Research Methods .................................................................................... 10
  1.5.3. Data Collection and Analysis .................................................................. 11
  1.5.4. Research Setting ....................................................................................... 12
  1.5.5. Summary ................................................................................................. 12
1.6. Validity Evaluation .............................................................................................. 14
1.7. Conclusions ......................................................................................................... 14
  1.7.1. Research Answers ..................................................................................... 15
1.8. Future Work ......................................................................................................... 16

Chapter 2 ..................................................................................................................... 19
2.1. Introduction ......................................................................................................... 19
2.2. Background ......................................................................................................... 20
  2.2.1. Suggestions for Trust Achievement ......................................................... 20
2.3. Research Methodology and Conduct ................................................................ 21
  2.3.1. Data Collection ......................................................................................... 22
  2.3.2. Data Analysis ........................................................................................... 23
  2.3.3. Validity of the Study ............................................................................... 28
2.4. Conclusions ......................................................................................................... 29

Chapter 3 ..................................................................................................................... 33
3.1. Introduction ......................................................................................................... 33
3.2. Background and Related Work ......................................................................... 34
  3.2.1. Agile Practices .......................................................................................... 34
  3.2.2. Global Software Engineering ................................................................. 34
  3.2.3. Agile Practices in Global Software Engineering ....................................... 35
  3.2.4. Related Work .......................................................................................... 35
  3.2.5. Motivations and Objectives .................................................................... 36
3.3. Research Method and Conduct ......................................................................... 36
  3.3.1. Research Questions ............................................................................... 36
  3.3.2. Search Strategy ...................................................................................... 36
  3.3.3. Data Sources ......................................................................................... 36
  3.3.4. Data Retrieval ........................................................................................ 37
  3.3.5. Inclusion Process ..................................................................................... 37
  3.3.6. Data Extraction and Synthesis ............................................................... 38
3.4. Results .................................................................................................................. 39
  3.4.1. Results of Literature Review ................................................................... 39
  3.4.2. Successful Applications ......................................................................... 40
  3.4.3. Limitation ................................................................................................ 45
3.5. Discussions .......................................................................................................... 45
3.6. Conclusions ......................................................................................................... 47

Chapter 4 ..................................................................................................................... 62
4.1. Introduction ......................................................................................................... 62
4.2. Related Work ...................................................................................................... 63
6.4.1. Results of Literature Review ................................................................. 107
6.4.2. Successful Applications ................................................................. 108
6.4.3. Limitations ....................................................................................... 115
Chapter 1

Introduction

1.1. Overview

Global Software Engineering (GSE) is rather a new concept that has appeared and evolved in the past two decades. It has received attention mainly due to its perceived benefits such as reducing the development time by benefiting from “follow-the-sun” software development [5][9], closeness to market, and accessing a large pool of skilled developers.

In contrast, many risks are also associated with GSE where people with potentially different cultural backgrounds and different social norms work together over a physical distance to solve some problems [7]. The distance introduces many new problems (e.g. lack of face-to-face meetings) that did not previously exist in traditional co-located software development.

Trust has been identified as an indicator of success or failure of partnerships, strategic alliances, and networks of firms [1][6]. It is also crucial for all business relationships as it enables more open communication, increased performance, higher quality deliverables, and greater satisfaction in the decision-making process [6]. However, distance hinders building trust among team members and sustaining it afterwards. The main reason could be lack of “teamness” feeling between team members since they do not meet [9].

The basic problems in GSE are related to communication, coordination, and control since traditional co-located mechanisms do not work well for distributed teams [1][3]. However, they can be potentially alleviated if trust between remote parties exists.

On the other hand, in striving for more efficient software development approaches, Agile values and principles were initially formulated by practitioners [23]. Agile software development aims at responding flexibly and quickly to changes in customers’ needs, and hence, encourages continuous requirements gathering, establishing close collaboration between customers and developers, and frequent informal face-to-face communications [25].

This research focuses on efficient software development approaches in particular in (globally) distributed settings. Thus a number of empirical studies have been carried with the purpose of investigating the dynamics of trust in GSE, identifying efficient Agile practices in different GSE contexts, and examining the applicability of existing Agile assessment tools.

The utilized research methods include systematic literature review and case study. As part of the latter, a variety of secondary data collection methods approaches are utilized such as semi-structured interviews, questionnaires, group discussions, and presentations.

Building and maintaining trust in GSE collaborations was realized to be crucial and the success factor for trust was recognized to be the “awareness” of particular GSE challenges. However, these challenges shall be communicated properly to all distributed team members and proper actions shall be taken to address them.

Trust achievement is crucial for efficient GSE collaborations regardless of the applied software development method. Although Agile promotes trust among team members, its values and principles were formulated without considering the distribution of development teams. Hence, combining Agile and GSE is challenging.
Besides, several successful implementations of Agile methods in different GSE contexts were reported in the research literature. However, despite utilizing two different methods for searching the literature (database searching and backward snowballing) the identified patterns were similar. Although the papers found through the research approaches were different, the most common Agile practices in GSE were “standup meetings” and “sprints/iterations” in both studies. The similarities found in the results may however indicate the stability of findings in the two systematic reviews on the investigated topic (i.e. Agile and GSE).

Nevertheless, the current literature reports “Agile” as a general term and addresses the most common team/organization setting as “distributed team” which are not sufficiently informative about neither the actual applied Agile practices nor the distribution setting. This motivated us to search for applicable methods of assessing or profiling Agility in software engineering. Hence, the applicability of existing Agile assessment tools in an industrial setting was examined, and one of the studied tools was found to be sufficiently applicable in the context of the case organization.

The existing research literature in the area contains several successful cases of combining Agile and GSE while practitioners and researchers do not seem to be yet consistent regarding their perception of Agile practices and how to document the context of empirical studies.

This thesis contributes to GSE area by visualizing the trust dynamics in the project’s life cycle and proposing useful practices for each state of building, maintaining, or improving trust. Furthermore, it summarizes the status of research in the area of Agile GSE, and takes initiative for bringing the Agile research and practice communities into common perceptions regarding profiling/assessing Agility.

The remainder of the chapter is organized as follows: Section 1.2 provides background information on GSE as well as Agile methods, and Section 1.3 discusses the research gaps and the research contributions. The research questions are stated and motivated in Section 1.4. To answer the research questions, the utilized research methods for each study are elaborated in Section 1.5. The general validity threats with the studies constituting the thesis are discussed in Section 1.6. Finally, Section 1.7 concludes the introduction and summarizes the outcomes of the individual studies, and Section 1.8 outlines the future research directions.

1.2. Background

The context of this study is mainly GSE and Agile software development methods, which are elaborated in this section.

1.2.1. Global Software Engineering

The major difference between (globally) distributed and co-located software development is recognized to be “distance” with three different dimensions as temporal, geographical, and socio-cultural [8]. Each dimension of distance amplifies or introduces some challenges related to the communication, coordination, and control activities in software development [8]. It implies that the traditional co-located mechanisms for software development are not necessarily effective when distances exist.

Nevertheless, the interest in distributed software development either within or across the national borders has grown in the past two decades [1][2][4] mainly due to its perceived benefits such as shortening the delivery time, closeness to market or customer, using local skilled people, saving development costs, and marketing benefits of globalized presence [27].

Geographically distributed software development teams characterize distributed software development, whereas globally distributed teams characterize global software development [9]. In this research, we have considered both as GSE, which is also known as Global Software Development (GSD), Distributed Software Development (DSD), or Globally Distributed Software Development (GDSD) in the research literature.

GSE collaborations can be configured in multiple settings. Organizations can seek solutions from providers within the country, or in other countries. Besides, the development can be distributed either within the organization (e.g. different sites), or to other organizations. The description of different GSE settings is inspired by [10], and presented as follows.
**Onshore Insourcing:** In this setting, software development is performed among two or more units/sites of the same organization (insourcing) located in the same country (onshore).

**Onshore Outsourcing:** In this setting, an organization provides software development services to another organization (outsourcing) which both are located in the same country (onshore).

**Offshore Insourcing:** This configuration is when the collaboration is within the same organization (insourcing), but in different sites located in different counties (offshore).

**Offshore Outsourcing:** In this configuration, an organization purchases software development services from international (offshore) external providers (outsourcing) i.e. both organizations and countries are different.

Table 1.1 visualizes the GSE configurations. It should be, however, noted that co-located software development is when team members are located in one site ideally next to each other.

<table>
<thead>
<tr>
<th>Countries</th>
<th>Different</th>
<th>Offshore insourcing</th>
<th>Offshore outsourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Same</td>
<td>Onshore insourcing</td>
<td>Onshore outsourcing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Same</td>
<td>Different</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Organizations</td>
<td></td>
</tr>
</tbody>
</table>

### 1.2.2. Agile Software Development

Plan-driven software development encourages detailed planning from the beginning of a project. Its most well known instantiation is the waterfall process [13], in which steps representing different software development disciplines (e.g. requirements engineering, architecture design, implementation, and quality assurance) are executed sequentially, or potentially with feedback cycles.

Due to micro level planning and lengthy documentation, the plan-driven approaches are often referred to as heavyweight, and the major challenge associated with them is the long development time that hinders considering any changes in the requirements. Thus, it is not well suited for settings in which customer or market demands change frequently and quickly.

To address this issue, incremental software development methods were proposed in 1957 [15]. Later, lightweight software development methods evolved and recognized as Agile methods in 2001 [25], when a group of practitioners came to value “individuals and interactions over processes and tools”, “working software over comprehensive documentation”, “customer collaboration over contract negotiation”, “responding to change over following a plan” in discovering more efficient ways to develop software [25][37].

The core of Agile software development is to produce software iteratively and incrementally in order to facilitate rapid and flexible response to changes [30][31]. Thus, tasks are divided into small increments with minimal planning (i.e. time-boxed iterative approach). An Agile iteration is typically from one to four weeks, and releasing new features or products usually takes several iterations. Nevertheless, the working software is considered as the primary measure of progress.

An Agile team is small (i.e. normally 5-9 people), cross-functional, self-organizing, and is involved through the full software development cycle. While no corporate hierarchical roles are assigned to any of the team members, they individually take responsibility to meet requirements of each iteration. In addition, each team must contain a customer representative who is appointed by stakeholders to act on their behalf [33].

In most Agile implementations, team members meet every day and share what they did yesterday, what they intend to do today, and what the difficulties are [36][38]. Besides, stakeholders and the customer representative get together at the end of the iteration to review progress and reprioritize the tasks.
Well-known Agile software development methods include Scrum, Extreme Programming (XP), Feature Driven Development (FDD), Agile Unified Process (AUP), Crystal Clear, Dynamic systems Development Method (DSDM), lean software development, and Kanban [34][35][36].

These methods support different aspects of software development. For example, DSDM covers the whole development life cycle, XP focuses on development practices, and Scrum emphasizes on management practices. However, according to the research literature (see Chapter 3), most methods have been tailored before being applied to specific GSE contexts.

Most Agile methods often utilize several tools and techniques (hereon we refer to these as Agile practices) for improvement of software quality and improving the Agility. The majority of these practices, however, are derived from XP’s four core values (i.e. communication, simplicity, feedback, courage) [33]. XP practices are: (1) planning game, (2) small releases, (3) system metaphor, (4) simple design, (5) testing, (6) refactoring, (7) pair programming, (8) collective code ownership, (9) continuous integration, (10) sustainable pace, (11) whole team, (12) coding standards, and (13) onsite customer. In literature, the terminology for reporting Agile practices is not consistent and practices are documented differently. For example, “sustainable pace” is also called “40-hour week”.

Despite the popularity of Agile methodologies, the scientifically based evidence to support their claims in achieving success and improving the quality is not sufficient [39]. For this purpose, some practitioners conducted a survey by which 55% of the respondents claimed successful implementation of Agile in 90-100% of cases [16]. Although such studies are useful, there is not yet sufficient evidence to conclude the success of Agile methods. Besides, Agile might not be efficient enough in large organizations and certain types of projects [38] due to its very definition (e.g. emphasis on small teams, collocation, etc.) that complicates its application on different software development contexts (e.g. distributed teams).

1.3. Research Gaps and Contributions

In uncovering approaches to achieve efficiency in software development and minimizing the costs, software organizations are increasingly investigating different ways such as outsourcing, offshoring, “near-shoring”, using open source software components, or Agile methodologies. However, all these different approaches can be classified in two groups of trends in software engineering. One is distribution (either within or across the national borders), and the other one is Agile software development methods. Furthermore, the interest in combining both trends has also grown in the past decade (see Chapter 3).

In spite of the rapid interest, it is not yet scientifically shown that either of the trends is efficient in comparison with more traditional software development methods. A major challenge in distributed contexts is to find effective mechanisms for controlling and coordinating the projects as well as communicating among dispersed teams [32]. Thus, the area of GSE is both promising and challenging. The recent research has sufficiently discussed the challenges with GSE, however, how to conquer or alleviate them is not yet well explored [28]. Although distribution results in new challenges in software development, most issues are actually the co-located problems, and “distance” only amplifies their severity [8].

Hence, one research direction is to study how to address the co-located software development issues in GSE settings. For example, “trust” related issues can also arise in co-located software development, and distance may only amplify them.

Another research direction is to study how new software development approaches (i.e. Agile methods) assist alleviating GSE specific challenges. Agile methods were formulated with the purpose of coping with limitations of traditional methods, but its practices do not take distribution of development teams into consideration, and hence, are primarily targeted at co-located teams. Thus, it is interesting to investigate how Agile can be tailored to alleviate GSE challenges and yet be loyal to its own values and principles.

Therefore, the major focus of the thesis is on combing Agile and global software engineering. However, the focus of each specific study included in the thesis is chosen based on the gaps that we identified through different literature reviews on Agile, GSE, or both. These gaps are discussed in depth in the relevant chapters.
In this chapter, we discuss the gaps in the existing empirical research related to GSE, which motivated individual studies constituting the thesis. Empirical studies in GSE are classified and the research gaps are summarized in [28]. The relevant areas for further investigation in the scope of this thesis are (1) studying different development methodologies, and (2) studying inter-organizational collaborations. Hence, our research particularly focuses on Agile methodologies in GSE contexts. However, both the trust achievement study (Chapter 2) and the Agile GSE literature reviews (Chapters 3, 4) may partially be relevant to inter-organizational collaborations as well as the Agility assessment tools study (Chapter 5).

Furthermore, a recent review of empirical research on Agile methodologies highlighted several areas that require further research [12]. The relevant items are that (1) the majority of current research is focused on a single method (e.g. XP), and (2) the experience of mature Agile organizations is rarely reported. Therefore, we have investigated the most common variations of Agile methods in the conducted studies (Chapters 2, 3, 4). In addition, the studied organization is experienced in Scrum (Chapter 5).

In addition to the previous studies [28][12], the results of our own systematic literature reviews on Agile GSE (presented in Chapters 3, 6) indicated the need for clear mechanisms of assessing or profiling Agility in software engineering. For this purpose, we conducted Study 4 (S4) in an industrial setting to examine the applicability of a set of existing tools for assessing Agile (reported in Chapter 5). Besides, due to the difficulties that we observed when applying the method for searching the research literature, we conducted Study 3 (S3), in which literature was explored differently and the results of two methods were compared (see Chapter 4).

The contributions of the thesis therefore can be summarized as investigating how to alleviate GSE-related challenges (C1), examining literature search approaches (C2), and examining the applicability of Agile assessment tools (C3). These major contributions are realized through the sub-contributions provided by each individual study included in the thesis. Study 1 (S1) discusses trust dynamics in GSE that addresses C1, and Study 2 (S2) is a systematic review of the current research literature on Agile GSE that also contributes to C1. S3 evaluates the two methods for searching literature, hence it contributes to C2, and finally S4 investigates the applicability of Agile tools which is related to C3. This is summarized in Table 1.2.

<table>
<thead>
<tr>
<th>Study</th>
<th>Chapter</th>
<th>Main Contribution</th>
<th>Addressed research gaps</th>
<th>Sub-contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>2</td>
<td>C1</td>
<td>* Inter-organizational collaborations</td>
<td>* Investigating trust dynamics in GSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* GSE challenges</td>
<td>* Best practices to build/maintain trust</td>
</tr>
<tr>
<td>S2</td>
<td>3</td>
<td>C1</td>
<td>* Inter-organizational collaborations</td>
<td>* Systematic review of most common Agile methods in all GSE settings</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* GSE challenges</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Agile methods in GSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Most common Agile methods</td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>4</td>
<td>C2</td>
<td>* Inter-organizational collaborations</td>
<td>* Systematic review of most common Agile methods in all GSE settings</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* GSE challenges</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Agile methods in GSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Most common Agile methods</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Efficient literature search methods</td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>5</td>
<td>C3</td>
<td>* Inter-organizational collaborations</td>
<td>* Evaluating the applicability of Agile assessment tools</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Most common Agile methods</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Experience of mature organization</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Assessing/profiling Agility</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Studying an experienced Agile company</td>
<td></td>
</tr>
</tbody>
</table>
In the following, the contents and contribution of each chapter is briefly presented, and their findings are explicitly linked to the contributions mentioned earlier.

Chapter 2 – Trust Dynamics in Global Software Engineering

Overview: Chapter 2 reports a qualitative empirical study that explores the dynamics of trust in GSE and provides industrial best practices to address the issues related to trust. Although trust is a key success factor in any software project, building and maintaining it is not straightforward particularly in (globally) distributed software projects. A literature review was firstly performed to study the dynamics of trust and how it is achieved in distributed teams. The collected information was also utilized in conducting interviews with eight project managers in six different software development organizations. The main purpose of the interviews was to further explore the trust relationship throughout the project life cycle and to identify the best practices to build and maintain trust in distributed settings. The outcome was a model to represent trust dynamics, and based on the findings, suggestions for the industry to achieve trust in GSE were provided. This research brings the awareness to practitioners of trust-related challenges when setting up GSE projects and hence to apply proper practices to mitigate them.

Contribution: In addition to proposing a model that illustrates trust dynamics in GSE, it provides industrial best practices to build, maintain, or improve trust. Hence, it is in alignment with C1, which is to discover approaches to mitigate GSE-related challenges in software development.

Chapter 3 – Global Software Engineering and Agile Practices: A Systematic Review

Overview: Chapter 3 presents a systematic literature review study with the purpose of capturing the status of applying Agile practices in GSE in the past decade. The combination of Agile and GSE is troublesome because Agile implementations seem to be inconsistent with GSE conditions. For example, Agile encourages co-located small teams whereas GSE introduces distance and distribution. The synthesis was made through classifying the papers into different categories (e.g. publication year, contribution type, research method). The existing research mainly consists of experience reports that have contributed by explaining the issues, specific solutions, and the lessons learned. However, the characteristics of project and the context were in many cases not sufficiently reported. This study depicts an overview of the status of the area, highlights the gaps, and helps to visualize the risks and benefits of Agile GSE.

Contribution: It describes a systematic review of successful implementation of Agile in GSE with the purpose of investigating under which circumstances the combination is effective. Therefore, it contributes to C1.

Chapter 4 – Systematic Literature Studies: Database Searches vs. Backward Snowballing

Overview: In the previous systematic review study in Chapter 3, we found the search method cumbersome. Therefore we conducted a separate study to evaluate the effect of search methods in the actual results and findings of such studies. Chapter 4 focuses on evaluating the two different search approaches for identifying the relevant papers, which are using search strings in a number of databases (the corresponding study is presented in Chapter 3) and snowballing (the corresponding study is reported in Chapter 6). The research questions and the process of data analysis were the same in both studies, and the comparisons were performed to find whether the same set of papers was found and if the included papers resulted in the same conclusions. However, regardless of the differences in the actual numbers and figures, similar pattern were identified in both studies, and hence, similar conclusions were drawn. The study also discusses the strengths and weaknesses of each method, which helps researchers in selecting the appropriate search method in systematic literature review studies.
**Contribution:** Similar to the previous study in Chapter 3, this research has explored the successful combination of Agile and GSE. This, however, was the secondary purpose (related to C1), and the primary purpose was to evaluate the influence of the search method in the actual results. So, it mainly contributes to C2.

**Chapter 5 – Investigating the Applicability of Agility Assessment Tools – A Case Study**

**Overview:** Throughout conducting the literature reviews, we realized that it is not specifically defined how much Agility would be sufficient in a particular situation, and it is hard to deduce from the papers exactly how Agile was implemented. Hence, Chapter 5 examines the applicability of existing tools for assessing or profiling Agility in software engineering. The tools were evaluated based on covered Agile areas and their comprehensiveness, and a set of them was selected as input to a case study with two software development teams. The assessment results provided by the tools were compared with the teams’ own perception of practicing Agility as well as their customers’ view. We realized that the studied tools do not assess the Agility similarly, and hence, they do not give the same scores to a specific team/organization. However, we recommend open discussions on the evaluation results with all team members and lead managers in order to prioritize the practices that are critical for the organization (e.g. are in alignment with the organizational goals). This implies a selective approach in adopting/improving Agility rather than encouraging being perfectly Agile. The results help both researchers and practitioners to gain awareness of the existing work in the area and to benefit from an analysis on the strengths and weaknesses of the studied tools.

**Contribution:** The focus of this study is on evaluating the applicability of the exiting commercial tools for assessing or profiling Agility in a software engineering context. This is in alignment with the third contribution of this thesis that is examining the applicability of Agile assessment tools. It should be noted that the tools do not explicitly take distribution of the teams into account and their questions are focused on the settings and practices that Agile methodologies demand. However, with some minor modifications in formulating the questions, it can also be fulfilled.

### 1.4. Research Questions

Table 1.3 provides an overview of the main research questions (RQ) and how they are connected to each contribution of this thesis. The main research questions are also linked to the research questions that are answered in the individual chapters.

The first contribution is the mitigation of the GSE challenges. The relevant research questions are RQ1 (trust achievement in GSE) and RQ2 (Agile practices in GSE) as stated in Table 1.3.

Chapter 2 answers three questions related to RQ1, which are RQ1.1 (trust evolution within distributed teams), RQ1.2 (best practices for building and maintaining trust), and RQ1.3 (trust achievement suggestions). This is summarized as follows.

<table>
<thead>
<tr>
<th>RQ1: How is trust achieved in GSE settings?</th>
</tr>
</thead>
<tbody>
<tr>
<td>• RQ1.1: How does trust evolve within distributed teams during the project life cycle?</td>
</tr>
<tr>
<td>• RQ1.2: What are the best practices the teams engage in for building and maintaining trust?</td>
</tr>
<tr>
<td>• RQ1.3: What are the suggestions for the industry to achieve trust in distributed collaborations?</td>
</tr>
</tbody>
</table>

Chapter 3 provides answers for RQ2 through answering RQ2.1 (the status of literature about Agile practices in GSE), and RQ2.2 (which Agile practices in which GSE settings, under which circumstances have been successfully applied).
RQ2: Are Agile practices applicable to GSE contexts according to the current research literature?

- RQ2.1: What is reported in the peer-reviewed research literature about Agile practices in GSE?
- RQ2.2: Which Agile practices in which GSE settings, under which circumstances have been successfully applied?

The second research contribution is about literature search methods. Hence, the main research question (RQ3) is about the effect of different search approaches on the results of systematic literature review studies. Chapter 4 is linked to RQ3, with discussing RQ3.1 (to what extent we find the same research papers using two different review approaches), and RQ3.2 (to what extent we come to the same conclusions).

RQ3: Does the method for searching the literature affect the results?

- RQ3.1: To what extent do we find the same research papers using two different review approaches?
- RQ3.2: To what extent do we come to the same conclusions using two different review approaches?

The third research contribution is directed towards Agile assessment. Hence, the main research question is formulated in RQ4 on the applicability of the existing tools for assessing Agility. Chapter 5 provides answers to this research question through investigating RQ4.1 (which commercial tools exist), RQ4.2 (the applicability of tools), and RQ4.3 (the differences in the given results by the tools).

RQ4: Are the existing tools sufficiently applicable?

- RQ4.1: Which commercial tools exist to evaluate the Agility of a team or an organization?
- RQ4.2: Are the existing tools applicable to assess Agility?
- RQ4.3: Do the existing tools give the same assessment results?
**Table 1.3. Mapping the research questions to the thesis contributions**

**Contribution 1: Research questions related to GSE challenges**

**RQ1: How is trust achieved in GSE settings?**

*Chapter 2: Trust Dynamics in Global Software Engineering*

- RQ1.1: How does trust evolve within distributed teams during the project life cycle?
- RQ1.2: What are the best practices the teams engage in for building and maintaining trust?
- RQ1.3: What are the suggestions for the industry to achieve trust in distributed collaborations?

**RQ2: Are Agile practices applicable to GSE contexts according to the current research literature?**

*Chapter 3: Global Software Engineering and Agile Practices – A Systematic Review*

- RQ2.1: What is reported in the peer-reviewed research literature about Agile practices in GSE?
- RQ2.2: Which Agile practices in which GSE settings, under which circumstances have been successfully applied?

**Contribution 2: Research questions related to literature search methods**

**RQ3: Does the method for searching the literature affect the results?**

*Chapter 4: Systematic Literature Studies: Database Searches vs. Backward Snowballing*

- RQ3.1: To what extent do we find the same research papers using two different review approaches?
- RQ3.2: To what extent do we come to the same conclusions using two different review approaches?

**Contribution 3: Research questions related to Agility assessment**

**RQ4: Are the existing tools sufficiently applicable?**

*Chapter 5: Investigating the Applicability of Agility Assessment Tools – A Case Study*

- RQ4.1: Which commercial tools exist to evaluate the Agility of a team or an organization?
- RQ4.2: Are the existing tools applicable to assess Agility?
- RQ4.3: Do the existing tools give the same assessment results?
Chapter 1 – Introduction

1.5. Research Methodology

Research methodologies assist researchers by providing guidelines to minimize bias and subjectivity in their investigations. In addition, they provide the link between research questions and the data to be collected in order to answer them [17]. Hence, the choice of a suitable methodology is critical that provides necessary data to enable researchers to answer the research questions.

Research methodology can be categorized in many different ways. One of the main categories of methodologies is empirical research that is evidenced based (i.e. findings are verified through observation and experience). Considering the stated research questions, this thesis focuses on empirical research methodologies. In the following, different types of methodologies that might be used in this research study are described.

1.5.1. Research Design

In order to ensure that the phenomenon is observed and influenced as intended, research designs provide different mechanisms [20][21]. They enable researchers to separate the effects of the treatments from other influences (e.g. uneven mix of people). Research designs are broadly categorized into two types: fixed and flexible. The research design is decided before the study is conducted in the fixed approach, whereas the design can be modified during data collection in the flexible approach [21].

1.5.1.1. Fixed Designs

In fixed designs treatment, control variables, and the procedure are specified in advance. Its different variations can be categorized [20] as (1) true experiments, (2) quasi-experiments, (3) single case experiments, (4) non-experimental fixed designs, and (5) systematic reviews. Among them, only systematic review fits into the scope of this thesis, which will be elaborated further in Section 1.5.2 and Section 1.5.3.

1.5.1.2. Flexible Designs

Flexible designs are used when the research is exploratory in nature, and the purpose is to construct a theory based on the perception of an individual or a group [20]. They can be divided [21] into (1) case studies, (2) grounded theory research, and (3) ethnographic studies. Case study and grounded theory are employed in the studies carried out in the thesis, and they will be explained in Section 1.5.2 and Section 1.5.3.

1.5.2. Research Methods

The most common approaches of empirical research in software engineering are case studies, surveys and experiments [20]. However, there are also other methods that can be utilized. In this chapter, we describe the commonly used methods focusing on those that particularly enable us to answer the posed research questions.

1.5.2.1. Surveys

Surveys are applicable when a sample must be studied with the intention of learning about a large population [17]. However, selecting the survey participants is challenging especially if the result are to be generalized. Data collection is performed through questionnaires or interviews, and statistical methods are used for analyzing the data.

1.5.2.2. Case Studies

A case study, usually investigates a phenomenon in its context [17][18]. The case study data is mostly qualitative, and can be collected in different ways depending on the needs of the study. However, the data collection can be direct such as interviews/observations, or indirect through document studies. This method is useful in exploratory studies where little is known about an area. The major difficulty with designing a case study is to separate a case from its context that might affect the generalizability of its findings [20][22].
1.5.2.3. Experiments

Experiments help to investigate the relationship between different factors through controlling related variables [17]. This approach is suitable to investigate aspects [22] such as (1) confirm theories, (2) confirm conventional wisdom, (3) evaluate the accuracy of models, (4) explore relationship, and (5) validate measures.

1.5.3. Data Collection and Analysis

A number of sub-methods have been used in the conducted studies, and are further described as follows.

1.5.3.1. Interviews

Interviews are conversations guided by an interview protocol and are considered one of the most important resources for data when conducting case studies [14]. The interview protocol can vary in the degree of structure, ranging from very structured (interviewee has to stick with research questions) over semi-structured (interviewee has a guide, but can change the course of the interview to follow interesting directions) to unstructured (rough definitions of topics to be covered). In the conducted case studies, we used semi-structured interviews to allow for some flexibility in the conversation. Unstructured interviews were not considered because interviews with some structures seem to be the most efficient way of eliciting information [11].

1.5.3.2. Group Discussions

Group discussion is an effective approach for gathering qualitative data. A group of participants and the researcher(s) sit together and discuss certain items depending on the needs of the case study. The researcher normally leads the discussions throughout the session. This method is efficient since it helps to collect data from several perspectives simultaneously, but it is challenging to find an occasion that all participants (especially industry practitioners) are available.

This approach has been used in S4 (Chapter 5) where the results of assessing the participating teams’ Agility provided by the tools were presented to them, and discussions were held to evaluate the applicability of the studied tools.

1.5.3.3. Systematic Literature Review

The primary goal of a systematic review study is to provide a fair evaluation of a research area by using a reliable, rigorous, and auditable methodology [19]. Such studies can be formulated as a systematic approach of interpreting and evaluating existing research in a particular research area, or related to a specific research question or a phenomenon.

1.5.3.4. Grounded Theory

Grounded theory studies the evolution of a theory to explain what is being observed [29]. It is useful to analyze an overwhelming amount of qualitative data [20]. The potential threat with this type of studies is the bias of the results, which occurs because the researcher often needs to have prior assumptions or theories when choosing the subjects. The detailed descriptions of the process can be found in the Chapter 2.

1.5.3.5. Statistical Analysis

Similar to grounded theory, descriptive statistics are a means of reducing the amount of data usually through visualizing the quantitative data [20]. We have used several types of diagrams in the conducted studies such as bubble-plots, box-plots, bar charts, etc.
1.5.4. Research Setting

Two primary settings used for data collection are presented as follows.

1.5.4.1. Anonymous Software Development Organization

The trust dynamics model that is presented in Chapter 2 has been extracted from the data collected through both literature review and semi-structured interviews with six software development organizations. The general information about the participating cases is provided in the corresponding chapter ensuring anonymity and confidentiality. The main advantage of collecting data from multiple industrial sources is that it enables researchers to draw more general conclusions and to pinpoint the trends in the studied area.

1.5.4.2. Softhouse Consulting

Softhouse Consulting is an independent IT consultancy company in Sweden, and currently is one of the leading Scandinavian suppliers of lean software development\(^1\). Softhouse was the industrial partner for the research presented in Chapter 5 that was conducted in cooperation with the development site located in Karlskrona, Blekinge, Sweden.

Softhouse has been helpful in formulating the research, facilitating the data collection, discussing the results, and promoting the changes according to the results. Two teams participated in the study, thus the results do not represent Softhouse as whole. Some information such as customer names and product-specific information is however not revealed for confidentiality purposes.

1.5.5. Summary

Chapter 2 presents a case study resulting in a trust dynamics model and best practices for trust achievement in GSE. Chapter 3 reports a systematic literature review on Agile practice in GSE. Chapter 4 illustrates a case study comparing two different methods of searching literature. Chapter 5 documents an industrial case study in which the applicability of Agile assessment tools are evaluated. Further information on each research study is given in the followings.

Chapter 2: Case Study

The case study elements with regard to the content of Chapter 2 are presented as follows.

<table>
<thead>
<tr>
<th>Strategy:</th>
<th>qualitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method:</td>
<td>semi-structured interviews, literature review, grounded theory</td>
</tr>
<tr>
<td>Phenomenon:</td>
<td>trust dynamics</td>
</tr>
<tr>
<td>Context:</td>
<td>global software engineering</td>
</tr>
<tr>
<td>Case:</td>
<td>software organizations involved in distributed software development</td>
</tr>
<tr>
<td>Participants:</td>
<td>eight project managers</td>
</tr>
</tbody>
</table>

Chapter 3: Systematic Literature Review

The guidelines provided by Kitchenham and Charters [24] for performing systematic reviews were considered in all steps undertaken in Chapter 3. The recommended steps include planning the review, conducting the review and reporting the results of the review [24].

The research started with defining a suitable scope, which was initially set to cover all Agile practices in all variations of GSE. Thus, the preliminary research questions were set and the keywords were identified. The initial keywords were searched in well-known databases such as ACM Portal and IEEE Xplore. Based on the search results, the research scope, the research questions, and the keywords were refined, search strings were reformulated, and searches were re-conducted. Moreover, the list of databases was expanded to collect as many relevant papers as possible. In parallel, a list of key papers was generated, which was used as a validation list to ensure the reliability and relevancy of the searches and to evaluate the search strings.

Chapter 4: Case Study

The case study elements with regard to the content of Chapter 4 are presented as follows.

<table>
<thead>
<tr>
<th>Chapter 4: Case Study</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strategy:</strong></td>
</tr>
<tr>
<td><strong>Method:</strong></td>
</tr>
<tr>
<td><strong>Phenomenon:</strong></td>
</tr>
<tr>
<td><strong>Context:</strong></td>
</tr>
<tr>
<td><strong>Case:</strong></td>
</tr>
<tr>
<td><strong>Participants:</strong></td>
</tr>
</tbody>
</table>

Chapter 5: Case Study

The case study elements with regard to the content of Chapter 5 are presented as follows.

<table>
<thead>
<tr>
<th>Chapter 5: Case Study</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strategy:</strong></td>
</tr>
<tr>
<td><strong>Method:</strong></td>
</tr>
<tr>
<td><strong>Phenomenon:</strong></td>
</tr>
<tr>
<td><strong>Context:</strong></td>
</tr>
<tr>
<td><strong>Case:</strong></td>
</tr>
<tr>
<td><strong>Participants:</strong></td>
</tr>
</tbody>
</table>

As described, this thesis employs multiple research methodologies. A summary of which research methodologies, collected data type and research setting is utilized in each chapter is presented in Table 1.4.
Table 1.4. Mapping the chapters to applied research methodologies

<table>
<thead>
<tr>
<th>Methods</th>
<th>Chapters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Qualitative</td>
<td>X</td>
</tr>
<tr>
<td>Quantitative</td>
<td>X</td>
</tr>
<tr>
<td>Case study</td>
<td>X</td>
</tr>
<tr>
<td>Survey</td>
<td>X</td>
</tr>
<tr>
<td>Experiment</td>
<td></td>
</tr>
<tr>
<td>Industry</td>
<td>X</td>
</tr>
<tr>
<td>Literature</td>
<td>X</td>
</tr>
<tr>
<td>Laboratory</td>
<td></td>
</tr>
</tbody>
</table>

1.6. Validity Evaluation

Validity threats related to each study are discussed in the corresponding chapter. A more general discussion on the threats is given in this section.

With the conducted case studies in Chapter 2 and Chapter 5, the main concern is related to the external validity because the number of cases is relatively small to the size of the population. However, in both studies, literature was also partially reviewed to find evidence supporting the findings.

On the other hand, the main concern with Chapter 3 and Chapter 4 is their reliability rather than generalizability. This is due to the fact that systematic reviews employ sound methodology for finding all relevant research papers that sufficiently assures their external validity. Nevertheless, the internal validity threats related to the reliability of each study are elaborated in the corresponding chapters as well as proper actions to address them.

1.7. Conclusions

In this thesis three contributions are made, namely the mitigation of the GSE challenges (C1), evaluating the literature search methods (C2), and investigating the applicability of Agile assessment tools (C3).

The first contribution is addressed in Chapter 2 and Chapter 3. A studied challenge is trust achievement in GSE collaborations in Chapter 2. In addition, the available research literature on the combination of the most common Agile practices and all GSE contexts are systematically reviewed to investigate the effects of Agile on GSE challenges.

To address the second contribution, Chapter 4 reports a study comparing two different search approaches in systematic review studies namely database searching and backward snowballing. In database searching, it is required to conduct searches in various databases, and usually each database requires its own search string formulation. On the other hand, backward snowballing starts with an initial set of papers and the list of papers increases by going backward through the list of references. This study compares the included set of papers in the analysis as well as the results and the derived conclusions.

Chapter 5 discusses the applicability of current Agile assessment tools, which contributes to assessing/profiling Agility in the software engineering discipline. A set of commercial tools was initially evaluated and three tools were chosen for further investigations in a case study with two Agile teams in a mature software organization.

During the course of this thesis, we have learned that generalizability can be hard to achieve in software engineering, mainly due to the fact that each software project is unique. The uniqueness is a result of differences in the context elements such as people’s, process’, product’s, and organizational features. The results of S2 (Chapter 3) confirm it, for example, we realized that the same issue is reported multiple times in literature while the solution is not the same. Although it indicates that a problem can be solved in many ways, it also highlights that one solution in GSE collaborations does not fit all situations.
Furthermore, the outcomes of S4 (Chapter 5) show that one tool for assessing/profiling Agility may not suit all teams/projects. This may be partly due to the contextual differences, and partly due to the flexibility of Agile methods that enables different teams practicing them differently.

However, case studies are useful means of gathering in depth information about a specific phenomenon and if the contextual information is sufficiently reported, the experience can be beneficial for other practitioners.

In literature, several successful cases are reported in which different Agile practices are applied in different settings of GSE. It should be noted that in most cases, the context is not properly documented and their best practices must be used with caution. However, it implies that modified Agile practices can mitigate GSE challenges such as fact-to-face communications.

1.7.1. Research Answers

Here, the results are summarized to provide answers to the posed research questions in the thesis.

RQ1: How is trust achieved in GSE settings?

Answer to RQ1.1: The state of trust evolves in GSE collaborations, from being built at the beginning to being maintained after its initial establishment. In some cases, trust might be injured which requires taking immediate actions to re-build it. The dynamics of trust are illustrated in a model in Chapter 2.

Answer to RQ1.2: However, for transitioning from/to each of the trust states in the model presented in Chapter 2, distributed teams need to employ certain practices for building and maintaining trust among themselves. The best practices for this purpose are collected from software organization involved in GSE.

Answer to RQ1.3: We recommend project managers to gain awareness of challenges in GSE prior to the collaboration. The awareness drives them to plan ahead for proper actions. However, in a critical situation, transparent and open communication with all people involved is required. The details can be read in the corresponding chapter.

RQ2: Are Agile practices applicable to GSE contexts according to the current research literature?

Answer to RQ2.1: In a conducted systematic review (reported in Chapter 3), the status of research in the area is summarized.

Answer to RQ2.2: The evidence is found in literature indicating successful accomplishment of GSE projects when Agile was utilized. The details of successful cases as well as GSE circumstances under which Agile is efficient can be seen in Chapter 3.

RQ3: Does the method for searching the literature affect the results?

Answer to RQ3.1: In the research presented in Chapter 4, it is shown that papers found through two different methods are different both in the number and the actual papers. In addition, the final set of papers, which was used in data analyses, was also different, although 27 papers were in common.

Answer to RQ3.2: Regardless of the differences in the actual numbers and figures, similar pattern were identified in both studies and hence similar conclusions were drawn. The detailed discussions on the observed similarities and differences can be accessed in Chapter 4 as well as the discussion on the strengths and weaknesses of each literature search approach.

RQ4: Are the existing tools sufficiently applicable?

Answer to RQ4.1: Chapter 5 reports a set of commercial tools that assess the Agility of a team or an organization, and discusses their the strengths and weaknesses.

Answer to RQ4.2: Three tools were examined in a case study and one of them was found to be applicable to the context of the studied teams. Comparing to the other two tools, the results achieved by the selected tool were more matched with the teams’ own perceptions of practicing Agile as well as their Scrum master’s and customers’ perceptions.

Answer to RQ4.3: The three tools examined in the case study gave different scores to the participating teams. A comprehensive discussion on the differences is presented in Chapter 5.
1.8. Future Work

The research began with the main focus on global software engineering. Therefore, trust-related issues in GSE were studied at the first place. Later, based on the discussions with the partners of the project, Agile methods were also added into the area of interest. Therefore, the systematic literature review was conducted to gain the knowledge in the area as well as further evaluating the successful combination of Agile and GSE. The findings were presented to the industrial partners, which led to the assessing/profiling Agility research study.

For future research, we would like to study how each Agile practice can alleviate GSE challenge(s). Since “distance” is the major difference between distributed and co-located software development, the research will be built on the challenges introduced/amplified due to different dimensions of distance (i.e. temporal, geographical, and socio-cultural) [8]. Each dimension of distance influences the communication, coordination, and control of the distributed projects [8]. Hence, GSE challenges can be categorized in nine groups (represented in Table 1.5). A research study can be conducted to examine how each group of issues can be addressed through different Agile practices.

Table 1.5. Nine groups of GSE challenges, adapted from [8]

<table>
<thead>
<tr>
<th>Dimension of Distance</th>
<th>Group A</th>
<th>Group D</th>
<th>Group G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coordination</td>
<td>Group B</td>
<td>Group E</td>
<td>Group H</td>
</tr>
<tr>
<td>Control</td>
<td>Group C</td>
<td>Group F</td>
<td>Group I</td>
</tr>
</tbody>
</table>

Furthermore, it would be interesting to study how the Agile assessment tool can be used in alignment with the organizational goals. For example, how different focus areas of lean development (e.g. reduce batch size, introduce cadence, delegate control, etc.) can be strengthened through improving different areas of Agility assessed in the survey found in Chapter 5.

References


Chapter 2

Trust Dynamics in Global Software Engineering

Abstract

Trust is one of the key factors that determines success or failure of any software project. However, achieving and maintaining trust in distributed software projects, when team members are geographically, temporally and culturally distant from each other, is a remarkable challenge. This paper explores the dynamics of trust and best practices performed in software organizations to address trust-related issues in global software engineering. Semi-structured interviews were conducted in six different distributed software development organizations and a resulting trust dynamics model is presented. Based on the findings, the paper also provides suggestions for the industry to achieve trust in distributed collaborations.

Keywords

Trust, Trust Building, Trust Maintenance, Global Software Engineering.

2.1. Introduction

Distributed teams comprising stakeholders from different national and organizational cultures, different geographic locations, and potentially different time zones characterize Global Software Engineering (GSE). These conditions have significant consequences on communication, coordination, and control [1]. Since software development depends on human interactions, addressing these challenges is critical for successful cross-border collaborations.

Mitigating the GSE challenges, however, is not a straightforward task. While frequent face-to-face communication in co-located teams supports achieving trust and a feeling of “teammness” among the remote colleagues, distance and cost-saving strategies in GSE often do not allow team members to travel between sites and meet [31]. In addition, different organizations may mean differences in the software processes [3], organizational standards, organizational cultures and policies, which might add additional difficulties to build and maintain cohesion and trust for the collaborating teams.

Given these constraints, distributed teams must rely on each other and find ways of working that tie them together. Trust is considered as the glue that holds the dispersed teams together and has been identified as an indicator of success or failure of distributed projects [2][28]. When trust exists, it enables more open communication among team members, which increases their performance and quality of the products at the end [2]. Team members have predictable behaviour and can therefore rely on each other to successfully accomplish the work [11].

Therefore, project managers have to seek strategies for addressing trust-related issues and engage distributed teams in the activities directed towards building, maintaining and improving trust, which we call in this paper as trust achievement. Although the significance of trust in the context of international organizations that exploit distributed software team is very well understood [28], the dynamics of trust in distributed teams requires deeper investigation for bringing useful suggestions to the project managers for trust achievement as well [12]. Moreover, a recent systematic literature review on the evidence in GSE-related research literature [37] identified that the amount of empirical studies in GSE is relatively small.
This paper explores the trust in GSE collaborations based on a qualitative empirical study. First, a literature review was performed to investigate the trust dynamics and how trust is achieved in distributed teams. Then, interviews were conducted in six different software organizations in order to further explore the trust relationship throughout the project life cycle and to identify the best practices to build and maintain trust among distributed teams.

The paper is organized as follows: Section 2.2 provides the background for this study. Section 2.3 presents the details of the qualitative study we conducted and discusses the findings. Finally, conclusions and future research suggestions are presented in Section 2.4.

### 2.2. Background

Trust is a multidimensional concept that can be explored at different levels such as within or among group(s), organization(s), or society [39]. It has been a topic of different disciplines such as philosophy, psychology, sociology, economics, and computer science [34]. Therefore, various trust definitions in different fields exist.

In this study, we consider the following definition: “the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control the other party” [23]. This implies that in a trust relationship there are two parties (trustor and trustee), a trust object, and a trust environment [34]. Furthermore, Rousseau et al. [32] stated that trust is not behaviour or a choice, but a psychological state that can cause or result from such actions. Therefore, trust has been viewed as a property of the relationship between parties, not as a property of the individuals [36].

Two major components of trust are recognized: the logically assessed component of trust that is called cognitive-based trust [14], and the social component known as affective-based trust. Cognitive-based trust is related to the rational characteristics of the trustees including reliability [24], responsibility [8], integrity, and competence [23]. Affective-based trust is related to the emotional and social skills of the trustees [4]. Building and maintaining trust in temporary work contexts depends more on the cognitive element of trust rather than the affective [25].

In globally distributed software projects, the main obstacles for trust achievement are reported as geographical, temporal, organizational, cultural and political differences [16][19], and distance [6].

Moe and Šmite [28] identified the reasons which result in lacking, incurring or losing trust as: poor socialization and socio-cultural fit, increased monitoring, inconsistent work practices, reduction of communication, unpredictable communication, lack of face-to-face meetings, conflict handling, lack of some of the characteristics required to have cognitive-based trust and poor language skills.

Casey and Richardson [7] highlighted the importance and impact of fear and its consequences on trust achievement. Huang and Trauth [17] reported the complexity of cultural understandings at different levels with respect to language issues, communication styles and work behaviours as trust achievement hindrances.

Lack of trust has severe impacts on performance of people, schedule, rework, and communications [28]. The major effects of lacking trust were identified to be the decrease in productivity, quality, information exchange and feedback, morale among the employees, and an increase in relationship conflicts. Therefore, trust is a prerequisite for the successful accomplishment of distributed software projects.

The following sub-section summarizes the current literature on the suggestions for trust achievement in GSE.

#### 2.2.1. Suggestions for Trust Achievement

Although the majority of the suggestions in the literature do not directly address trust, they implicitly improve trust building, maintenance or both.
2.2.1.1. Building Trust

Milewski et al. [27] proposed a bridging technique, in which one “bridge” location facilitates the collaboration and coordination across other locations. Mikawa et al. [26] suggested that open recognition of cultural differences and intentional strengthening of social ties among team members is important in distributed software teams.

Brannen et al. [5] observed that bicultural people (who have deeply internalized more than one cultural profile) are helpful in intercultural collaboration, communication, and trust building. Dual identity immigrant managers are also reported to be effective in collaboration and trust building [22].

2.2.1.2. Maintaining Trust

A simulation model for improvements in GSE and a sub-model for trust improvement are suggested in [35]. The model combines the system dynamics paradigm with the discrete-event paradigm.

In [33], a “Shared Project Context” model is explained to address the trust-related issues. And in [3], liaisons technique is proposed. The liaisons are engineers who moved to a remote office for a short period of time and their responsibility is to meet the developers, learn the system, help to complete the requirements and specifications, and communicate this information back to the development staff at their home office.

2.2.1.3. Building and Maintaining Trust

Kanawattanachai and Yoo [20] examined the dynamic nature of trust and the differences between high- and low-performing virtual teams, whose members are spread in different locations and work remotely. After observing the changing patterns in cognitive- and affective-based trust over time (early, middle, and late stages of project), it was concluded that high-performing teams were better at developing and maintaining trust and virtual teams relied more on a cognitive than an affective element of trust.

The results of an empirical study on software outsourcing relationships [2] show that cultural understanding, creditability, capabilities, pilot project performance, personal visits, and investment are important factors in building trust. For maintaining trust, in addition to these factors, communication, contract conformance, quality, timely delivery, development processes, managing expectations, personal relationships, and performance are reported as being significant factors.

In [14], the criticality of the three components of trust (ability, integrity, and benevolence) at each life cycle stage for a virtual team (i.e. team establishment, inception, organization, transition, and accomplishment of the task) were investigated. As a result, a set of action steps that shall be taken by the managers and the team leaders (such as how to choose team members or proper team building activities or to give support to team members) were mapped to each stage.

The literature shows an increasing number of studies, which have been conducted to understand trust achievement in GSE. However, the dynamics of trust and the industrial practices for establishing and maintaining trust in software organizations have not been deeply explored yet. In the following section, we discuss the results of a qualitative study we performed by conducting interviews in software organizations to investigate further trust-related practices and the dynamics of trust in their collaborations.

2.3. Research Methodology and Conduct

The major aims of this qualitative study were to understand the dynamics of trust in GSE and to shed light onto best practices to provide suggestions to industry on how to achieve trust in their collaborations. Our research questions were:

RQ1: How does trust evolve within distributed teams during the project life cycle?
RQ2: What are the best practices the teams engage in for building and maintaining trust?
RQ3: What are the suggestions for the industry to achieve trust in distributed collaborations?

In order to answer these questions, this research was designed as an exploratory study. The following sub-sections discuss the data collection and analyses steps.
2.3.1. Data Collection

In order to collect data, we first prepared a questionnaire based on the findings of the current literature review (see Section 2.2) on the causes of lacking, injuring or losing trust in GSE as well as the suggestions for trust achievement.

Then, we conducted semi-structured interviews\(^1\) (10 one-hour interviews) with project managers from six different software organizations (involved in eight different GSE projects) to explore further the dynamics of trust as well as the best practices in the industrial settings.

We selected the interviewees to represent different nationalities (Malaysia, Iran, Serbia, Sweden, and South Africa) under the constraint of the availability of participants. Furthermore, it was critical to include different cultures in this study to be able to observe the different trust building and maintaining behaviours since trust is very much dependent on people’s actions and perceptions that can be influenced by their cultural backgrounds.

In addition, we aimed at covering different types of business relationships in our case projects. Three projects were offshore insourcing and four were offshore outsourcing projects. Only one project was an onshore outsourcing project and none were onshore insourcing projects (see Table 2.2).

Some of the interviews were conducted via Skype and some through meeting in person depending on the distance and the interviewed manager’s preference. We used a qualitative research analysis tool, called NVivo 8\(^2\) to store and analyze the collected data.

Table 2.1 summarizes the information regarding the case organizations, the case projects and the performed activities by the teams located at different locations. Even though we provide all the locations involved in the case projects, we conducted the interviews so that at least one trust relationship could be captured and analyzed. The projects and the involved parties, for which we could have collected data, are shown in italic in the table. Detailed information about the interviews can be found in [18]. We cannot provide the names and further information regarding the organizations and projects due to confidentiality purposes. Instead, we use acronyms A, B, C, D, E and F to represent different organizations and numbers to represent different projects in which trustor and trustee teams collaborated.

<table>
<thead>
<tr>
<th>Project</th>
<th>Interviews</th>
<th>Investigated locations</th>
<th>Other locations</th>
<th>Investigated locations</th>
<th>Other locations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Country A Ds Dv T M</td>
<td>Country A Ds Dv T M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>Malaysia (\checkmark) (\checkmark) (\checkmark) (\checkmark)</td>
<td>Iran (\checkmark) (\checkmark)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>USA (\checkmark) (\checkmark)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>1</td>
<td>Sweden (\checkmark) (\checkmark) (\checkmark) (\checkmark)</td>
<td>Sweden (\checkmark)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2</td>
<td>1</td>
<td>Sweden (\checkmark) (\checkmark) (\checkmark) (\checkmark)</td>
<td>Ukraine (\checkmark) (\checkmark) (\checkmark)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>Serbia (\checkmark) (\checkmark) (\checkmark) (\checkmark)</td>
<td>Sweden (\checkmark) (\checkmark) (\checkmark) (\checkmark)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>1</td>
<td>Sweden (\checkmark) (\checkmark) (\checkmark) (\checkmark)</td>
<td>China (\checkmark) (\checkmark) (\checkmark) (\checkmark)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>1</td>
<td>Sweden (\checkmark) (\checkmark) (\checkmark) (\checkmark)</td>
<td>China (\checkmark) (\checkmark) (\checkmark) (\checkmark)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>China (\checkmark) (\checkmark) (\checkmark) (\checkmark)</td>
<td>Sweden (\checkmark) (\checkmark) (\checkmark) (\checkmark)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>South Africa (\checkmark) (\checkmark) (\checkmark)</td>
<td>France (\checkmark) (\checkmark) (\checkmark)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Romania (\checkmark) (\checkmark) (\checkmark)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>India (\checkmark) (\checkmark) (\checkmark)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>Sweden (\checkmark) (\checkmark)</td>
<td>Hungary (\checkmark) (\checkmark) (\checkmark)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A: Analysis, Ds: Design, Dv: Development, T: Test, M: Maintenance

\(^1\) A semi-structured interview is flexible and allows new questions to be brought up during the interview as a result of what the interviewee says [10].
Table 2.2 represents the information about the types of business relationships in each project. The given classification is inspired from [30].

We find it important to differentiate two major types of work relation for our further discussions in this paper. Among the studied organizations some projects formed co-located teams working on a separate phase or task independently. Others utilized virtual teams that consisted of distributed team members working jointly. The case overview is presented in Table 2.3.

<table>
<thead>
<tr>
<th>Different countries</th>
<th>Offshore insourcing</th>
<th>Offshore outsourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1, D2, F</td>
<td>A, B2, C, E</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Same country</th>
<th>Onshore insourcing</th>
<th>Onshore outsourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same organization</td>
<td>Onshore outsourcing</td>
<td></td>
</tr>
<tr>
<td>Different organization</td>
<td>B1</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2. Case overview: business relationships

<table>
<thead>
<tr>
<th>Projects</th>
<th>A, D1, D2, E</th>
<th>B1, B2, C, F</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Performance</th>
<th>Joint</th>
<th>Independent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teams</td>
<td>One virtual team</td>
<td>Several distributed teams</td>
</tr>
</tbody>
</table>

Table 2.3. Case overview: distributed project organization

2.3.2. Data Analysis

We analyzed the collected data to investigate the trust dynamics (how transition among trust states occurs and why) among distributed teams in each project life cycle and to identify industrial best practices for building and maintaining trust.

Data analyses were performed using Grounded Theory\(^2\) (GT) through applying open, axial, and selective coding techniques [38]. The resulting codes were re-checked for consistency and clearness before proceeding further for constructing the final outcome of this study.

Data analysis started with an open coding [38]. Interview text was reviewed to identify sentences about causes of lacking/losing trust, and related practices. The text was labelled with proper keywords. Similar codes were grouped together under a more general concept. Later, these concepts were grouped into categories. The following example explains how GT was used in data analysis.

**Interview Transcript X:** “Emails are used mostly because of language issues.”

**Interview Transcript Y:** “The mostly used communication method is IRC chatting. This method is also a preferred one, since it is synchronous and still enables both sides to avoid potential language issues and misunderstandings.”

The first case addresses one of the identified causes in the literature. Therefore, it was coded as “Linguistic Differences”. Furthermore, the applied practice was stated as “Email”. The second case addresses the same cause, but the practice is “IRC chatting”. Therefore, this statement is coded as “Chatting”.

---

\(^2\) By GT, the actual data of the real world is examined and analyzed in order to draw grounded theories [13]. GT suits well for exploratory investigations when there is no prior knowledge of a part of the reality or a phenomenon and no preconceived hypothesis [10].
In the next step, we grouped “Email” and “Chatting” into “Written Communication”. Later, “Written Communication” and similar concepts grouped in a more general category named “Practices”. Then, “Linguistic Differences” was grouped with other causes of lacking/losing trust and their consequences in a general category of “Threats”. Hence, the threat-practice relationship was recognized.

In the following sections, we present and discuss the results of the analyses on the collected data. First, we discuss the trust state transitions during the life cycle of each project. Then, based on these states, we present a trust dynamics model for GSE projects life cycle. Finally, we discuss the identified best practices in relation to the trust life cycle.

2.3.2.1. Trust State Transitions

For the following discussions on the trust state transitions within each project, the icon  demonstrates the state of trust,  represents the state of distrust [21] and  represents the state of neither trust nor distrust.

For each project, the first location in the shown relationship represents the trustor organization and the second – the trustee. The trustor is the product owner and the trustee is the team, which the trustor chose to collaborate with for that particular project. We investigated the trust relationship considering the perspective of our interviewees from either the trustor or the trustee teams.

Project A. Malaysia (trustor)  Iran (trustee):   

The virtual team including members located in Iran and Malaysia started their collaboration with a strong initial trust. The reason for the strong initial trust was stated to be the fact that many members of both teams had worked in the same co-located team previously. This initial trust in return facilitated effective communication among teams. Some practices were also planned at the beginning of the collaboration and performed during the project to maintain and improve trust (see the best practices in Section 2.3.2.3). Moreover, the progress was communicated daily and members of the virtual team were in contact via chat during the overlapping working hours and discussed the project-related issues. The common language spoken in Iran and in Malaysia helped in making the communication easier. The trust was maintained and improved until the end of the project.

In this case organization, we interviewed two different project managers from the trustor organization for the same project. They both had similar applied practices, which points an organizational awareness about the significance of trust and joint decisions for trust achievement.

Project B1. Sweden (trustor)  Sweden (trustee):     

For this project, we interviewed the trustee team. Since these teams were collaborating for the first time, the initial trust state was “neither trust nor distrust”. However, in the past, the trustee team collaborated in another distributed project with another trustor team, which had been a failure. Therefore, they previously had a negative experience in such distributed collaborations.

During this project, although a number of practices were performed to build and maintain trust among the teams, the teams could not build trust to the end of the project. The reasons were stated as the requirements and quality expectations were not well negotiated early among the teams. As a result, even though the final product was delivered with high quality according to the trustee, it did not satisfy all of the expectations of the trustor and thus, the trust was lost.

Project B2. Sweden (trustor)  Ukraine (trustee):     

This case project was also the first collaboration between the trustor and the trustee teams. Therefore, the initial trust state was “neither trust not distrust”. During the project, the Swedish team lost trust in the Ukrainian team in performing tests and verifying their work before delivery since the final product was delivered with many defects. However, later, the Swedish team continued to work with the Ukrainian team by changing the expectations, which was to still delegate all development responsibilities to the Ukrainian team, but re-test their work in Sweden. It was much cheaper to outsource the development to Ukraine and re-test the final product rather than developing and testing the product in Sweden.

The commonality of these two projects was the use of too few practices for addressing trust challenges. Although in Project B1 more practices were implemented than Project B2, the trustee could not meet most of the expectations of the trustor and the trust was totally lost and the collaboration terminated.
Project C. Sweden (trustor) ⇔ Serbia (trustee): ⬇️ ⬝ ➟ ⬝

Since there was no prior experience of working together in this collaboration, the Swedish team evaluated the trustworthiness of the Serbian team based on their expertise. Therefore, there was no strong trust state at the beginning. During the execution of the project, the Serbian team showed high performance and were able to meet deadlines. Furthermore, they maintained frequent informal communications with the Swedish team. The main success factor was stated to be the effective and frequent communication among distributed teams along with facilitating informal knowledge sharing. Instant message tools were used to decrease the delays in communication. This also increased the frequency of communication. In addition, they logged and kept track of the history of text messages for traceability and conflict resolution purposes in future. The collaboration ended in a trust state.

Project D1. Sweden (trustor) ⇔ China (trustee): ⬝ ➟ ⬝

The collaborating virtual teams in this project were offshore locations that belong to the same organization. The teams started with an initial state of “trust”. During the life cycle, exchanging team members and meeting schedule and quality expectations maintained trust. Furthermore, they planned for frequent face-to-face meetings and travelling between the sites in advance. In critical situations with high face-to-face interaction demands, key team members from Chinese team travelled to Sweden and worked together.

Project D2. Sweden (trustor) ⇔ China (trustee): ⬝ ➟ ⬝

In this project, we interviewed the project managers of both the trustor and the trustee teams. The offshore locations in this project also belong to the same organization. The teams started with an initial state of “trust”. Daily short informal meetings through conference calls were conducted to exchange information and to communicate the status of the project.

In both projects in this organization, starting with trust state and performing many practices to maintain and improve trust helped complete the projects with success and in a trust state.

Project E. South Africa (trustor) ⇔ France (trustee): ⬝ ➟ ⬝

In this relationship, the status of the initial trust was “neither trust nor distrust”. The South African team relied on the technical competence of the French team to start the collaboration. This organization was experienced in distributed projects. The interviewee was very well aware of GSE challenges and trust specific problems. The activities were planned well and the tasks were distributed among the locations. Task dependencies between the teams were minimized while partially dependent tasks were assigned to the teams separated by a small temporal distance. Moreover, the South African team clearly set the quality expectations and asked the French team to use specific standards and shared templates. They were able to build and maintain trust throughout the project life cycle despite the challenges of task distribution within different teams.

Project F. Sweden (trustor) ⇔ Hungary (trustee): ⬝ ➟ ⬝

The Swedish and Hungarian teams worked for the same organization in the past. Therefore, Swedish team initially trusted the other team from the beginning. During the project, the teams were able to maintain trust by regularly negotiating each other’s expectations and keeping promises.

After analyzing the trust state transitions in the case organizations, we further investigated the general dynamics of trust by exploring the patterns in the cases. The results are presented in the next section.

2.3.2.2. Trust Dynamics in the Life Cycle

We used the concepts and components (the trustor, trustee, trust object, and trust environment) of Schultz’s situational trust model [34] in order to model the general trust dynamics within the distributed project life cycles by exploring the case projects (see Figure 2.1).

There are two phases of trust in distributed collaborations: the initial trust building phase and trust evolution phase. The initial steps in the diagram can be viewed as initial trust building phase, which ends when “trust” state is achieved after the expectations are agreed. The first phase is called as static since the project starts after this phase when an acceptable level of trust is achieved.
During the initial trust building phase, there is an interaction between the trustor and the trustee. The initial trust state of the trustor is based on the previous situation specific interactions with the trustee. In the case of no previous interaction, the trustor relies upon former experiences and/or evaluates the trustworthiness of the trustee. Therefore, the initial trust state can also be a state of no strong trust or distrust. Based on this knowledge, the trustor sets the expectations from the trustee and the trustee perceives these expectations.

When an acceptable level of trust is built (based on the expectations) the collaboration starts, and this “trust” state initiates the dynamic phase of trust evolution. During the project life cycle, the trust state might continue to be maintained, injured and rebuilt, or totally lost. As long as the actual behaviour of trustee is matching with the agreed expectations, trust is maintained. The resulting trust state (“trust”, “distrust”, or “injured trust”) is based upon the trustor’s perception of and experience with the trustee, the trust object, and the environment.

![Figure 2.1. Trust dynamics in the project’s life cycle](image)

The resulting trust state can be observed as “initial trust” for the future collaboration possibilities. When the previous collaboration completed in a “trust” state, in the new collaboration the trust is usually built and maintained easier. On the other hand, “injured trust” (trust is partially lost) or “distrust” (trust is totally lost) states might terminate any further collaboration. In such a situation, the trustor party makes a decision whether changing the expectations (the trust object) or the environment might help to “rebuild” the trust. (see Section 2.3.2.1 for more details on trust states transitions in case organizations).

2.3.2.3. Best Practices for Trust Achievement

In this section, we present the identified best practices for trust achievement. For each practice, information on the source organization along with a brief elaboration is provided. Recommendations in each category are ranked considering their popularity, i.e. how often the practice was mentioned by the interviewees. Hence, the ranks of the following recommendations represent their popularity among case organizations. Table 2.4 maps the identified practices to the investigated case projects.
Table 2.4. Map between recommendations and organizations

<table>
<thead>
<tr>
<th>Info.</th>
<th>Recommendation Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>√</td>
</tr>
<tr>
<td>B1</td>
<td>√</td>
</tr>
<tr>
<td>B2</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>√</td>
</tr>
<tr>
<td>D1</td>
<td>√</td>
</tr>
<tr>
<td>D2</td>
<td>√</td>
</tr>
<tr>
<td>E</td>
<td>√</td>
</tr>
<tr>
<td>F</td>
<td>√</td>
</tr>
</tbody>
</table>

Recommendation 1 Organizations: A, B, C, D, E, F

Plan the communication and regular meetings in advance

Planned communication prescribes defining media, contacts, timelines, rules and regulations. Regular meetings can be held either face-to-face or over (video) conference calls. These increase the predictability and ensure frequency of communication.

Recommendation 2 Organizations: A, B, C, D, E, F

Prevent misunderstandings

The frequency of misunderstandings during communication of the distributed teams is high. It is stated to be critical to identify the major causes and to address them early in the life cycle. For example, one significant reason was identified to be poor language skills. During the interviews one of the comments to overcome this issue was to utilize written rather than oral communication especially when the teams do not have very good level of the language used for communication.

Recommendation 3 Organizations: A, B, C, D, E, F

Encourage informal communication

Any kind of informal communication may compensate the lack of socialization in GSE. It can be achieved through unplanned chat or calls.

Recommendation 4 Organizations: A, C, D, E, F

Use common work processes, shared templates and standards

Teams working on the shared tasks shall agree upon the work processes, otherwise team members usually experience confusion and misunderstandings, for example, when integrating the work of different parties.

Recommendation 5 Organizations: A, C, D, E

Minimize delays in communications and in conflict resolution

Utilizing synchronous communication methods together with distributing dependent tasks among close time zone locations shortens response time. Moreover, it is crucial to communicate the issues and conflicts immediately to resolve the conflicts as early as possible.

Recommendation 6 Organizations: A, B, D

Collect regular status reports from each team member

Status reports help project managers to monitor the performance of the team members, to track the project progress and take timely actions, thus, avoiding injuring trust due to time and cost overruns. This practice also helps in building cognitive-based trust and avoiding over-control of the remote team members.
Chapter 2 – Trust Dynamics in Global Software Engineering

Recommendation 7 Organizations: C, D, E

Make the communications traceable

Keeping the history of communications provides the possibility to review the communications later if a conflict happens. Furthermore, tracking the decisions for a specific matter becomes easier.

Recommendation 8 Organizations: A, D, E

Cooperate closely in the case of an urgent need

In few occasions such as high task dependencies or in solving severe conflict issues, face-to-face and close cooperation is highly recommended. This can be achieved, for example, through staff exchange.

Recommendation 9 Organizations: A, D

Gain cultural awareness

Before and during cooperation with remote sites, it is crucial to gain awareness of cultural differences either through experience or training.

Recommendation 10 Organizations: A, E

Be available for your remote colleagues

Availability is an important factor for the team’s cohesion. It reduces delays in communication and improves the links among remote team members.

Recommendation 11 Organizations: A, D

Exchange team members across locations

This recommendation alleviates the lack of face-to-face meetings through socialization. This stimulates active information exchange between teams during and, most importantly, after the co-location.

Recommendation 12 Organizations: A, D

Encourage sharing of best practices among distributed teams

Encouraging team members to share best practices increases the “teamness” feeling among them and helps to achieve the shared goal.

Recommendation 13 Organizations: A, D

Encourage use of video in communication

The interview results suggest that video can partially compensate the absence of meeting in person and significantly improves communication.

2.3.3. Validity of the Study

Below, we discuss the validity threats regarding reliability and generalizability of this research and what we did to overcome.

Internal Validity: Internal validity aims at ensuring that the collected data enables the researchers to draw valid conclusions [10]. Therefore, the transcript of each interview was prepared immediately after the interview to minimize the risk of forgetting some parts of unwritten information since the interviews were not recorded. Furthermore, the transcription document was sent back to interviewees to confirm the content.

It should be noted that there is not much evidence in the current research literature to believe that the results of face-to-face interviews vary from the Skype-based. Therefore, conducting interviews in two different ways (face-to-face and over Skype) has not affected the quality and reliability of the results of this study.

However, triangulation technique (a method that compares three or more types of independent perspectives on a given aspect of the research process (methodology, data, etc.) in order to improve the accuracy of findings) [15] was applied to ensure the internal validity of the research. The triangulations used in this study were data and investigator triangulations.
Data Triangulation: The data was collected during the interviews with managers who have different experience and expertise. The interviews were designed in a way to avoid directly relating the questions to trust issues. A small sample of participants from the senior developers working at Swedish organizations and senior software engineering students studying at Blekinge Institute of Technology also checked the questions before conducting real interviews. Hence, the content was refined until we agreed that questions are clear enough for interviewees.

Investigator Triangulation: In data collection and data analyses, more than one researcher was involved in performing and validating the work. Other researchers reviewed the findings from each researcher and comparison was made to ensure that their conclusions were similar.

One limitation of this study was that the data could not be collected from both trustee and trustor parties involved in the case projects due to availability reasons. However, we believe that this would not significantly affect the reliability of the discussions and contributions of this study. First, the final trust state is associated with the outcome of the business relationship, thus trust should not be subjectively misperceived and both trustor and trustee are expected to have the same perception about the final trust state. Second, the identified practices were performed during the collaboration of both parties and therefore should not be different.

External Validity: External validity defines to what extent findings from the study can be generalized to and across populations of persons, settings, and time [10]. Hence, proper actions to overcome relevant threats were considered in the design of this study.

This research aimed at finding practices that would apply to different types of collaborations of distributed teams. Project managers working in different companies collaborating in different ways with other teams to develop different types of software products were interviewed (see Section 2.3.1 for more details). Moreover, in order to increase cultural diversity of the population, we interviewed the managers from the organizations located at different parts of the world (Asia, Africa, and Europe). Even though the details of projects are not available, the discussions presented in this study can be generalized for similar contexts e.g. offshore development.

There is not much reason to believe that the best practices can be generalized over time. The technology is evolving and new tools will be introduced to support best practices. However, we believe that the dynamics of trust will still be valid over time.

2.4. Conclusions

This study explored the dynamics of trust and best managerial practices to overcome the challenges of building or maintaining trust during the collaboration of globally distributed teams.

Based on our findings, we suggest managers who start a distributed collaboration to consider the following factors.

Trust Dynamics: The trust dynamics model developed in this study revealed that initial trust building is a static process in which the trustworthiness of the trustee is evaluated and the expectations are negotiated. One outcome of this is that if the expectations were not clear and well-set from the beginning, the practices conducted in the following dynamic phase to achieve trust when the project starts, do not help much since there is a high risk that trust might be injured (the behaviour of the trustee not match with the expectations of the trustor due to this unclearness). If this situation is avoided from the beginning, then it is critical that project managers should plan and engage the team members in practices towards maintaining and improving trust.

The Type of Business Relationship: Our observations indicate that business relationship has a significant effect in determining whether the project will start with a strong initial trust. In the investigated organizations, teams that were formed by members of the same organization shared corporate identity and thus implied the trustworthiness of the trustee. On the contrary, lack of previous collaboration experience and shared organizational background hindered strong trust at the beginning. This may motivate the managers to invest more in trust and cognition achievement activities.

The Role of Management: The best practices presented in Section 2.3.2.3 highlight the role of managerial actions in the trust relationship between distributed teams. A success factor for trust was recognized to be the “awareness” of the particular challenges in GSE. Especially, good communication management, which addresses these challenges, is essential.
An important observation in this study is that all of the managers participated to our qualitative study expressed a great interest in this study and to our findings. They also mentioned the need to further investigate trust-related issues as well as ways to achieve trust in GSE to be able to learn from others’ experiences.

As the future work, we suggest conducting a similar study with the software developers to explore their viewpoints and awareness in comparison to project managers.

It would also be interesting to investigate further how different collaboration settings such as “near-shoring” and “far-shoring” would affect the trust dynamics and trust building and maintenance practices.

Acknowledgements

We would like to thank Branislav Zlatkovic for his great help and participation in the data collection and analysis phases of this study. We also thank our interviewees for giving us a part of their precious time, the valuable information and useful feedbacks. Our great gratitude goes to Professor Claes Wohlin for his review, guidance and useful suggestions.

References


Chapter 3

Global Software Engineering and Agile Practices: A Systematic Review

Abstract

Agile practices have received attention from industry as an alternative to plan-driven software development approaches. Agile encourages e.g. small self-organized co-located teams, whilst global software engineering (GSE) implies distribution across cultural, temporal, and geographical boundaries. Hence, combining them is a challenge. A systematic review was conducted to capture the status of combining Agility with GSE. The results were limited to peer-reviewed conference papers or journal articles, published between 1999 and 2009. The synthesis was made through classifying the papers into different categories (e.g. publication year, contribution type, research method). At the end, 81 papers were judged as primary for further analysis. The distribution of papers over the years indicated that GSE and Agile in combination has received more attention in the last five years. However, the majority of the existing research is industrial experience reports, in which Agile practices were modified with respect to the context and situational requirements. The emergent need in this research area is suggested to be developing a framework that considers various factors from different perspectives when incorporating Agile in GSE. Practitioners may use it as a decision-making basis in early phases of software development.

Keywords

Agile Practices, Global Software Engineering, Distributed Software Development, Systematic Review.

3.1. Introduction

Distributed teams consisting of stakeholders from different national and organizational cultures, different geographic locations, and potentially different time zones characterize global software engineering. These characteristics have significant effects on communication, coordination, and control, and mitigating the effects is a challenge [26].

In comparison with plan-driven software development approaches, Agile methods are more flexible when it comes to taking requirements’ changes into consideration in all phases of software development [11]. They emphasize extensive collaboration between customers and developers, and encourage small self-organized co-located teams [19].

Although mitigating the GSE challenges by themselves is not a straightforward task, combining Agile practices with a global or distributed context complicates things even further. Frequent face-to-face communication among co-located team members improves a feeling of “teamness” and builds trust [7], whilst distance in GSE implies a different way of working, organizational standards, organizational cultures and policies, which may decrease the team’s cohesion.
However, (globally) distributed Agile has attracted attention due to its potential associated benefits such as shorter time to market, reduced development cost, and managing late requirements’ changes. This indicates the need for investigating the experiences reported in the current research literature to determine how Agile practices can be efficiently applied in (globally) distributed projects. Although several studies have reported successful integration of Agile and GSE (e.g. [25][9]), a thorough analysis of the studies to reveal the applicability of the reported experiences and best practices in different organizational settings and project demands is yet unexplored.

This research primarily aims at extending the systematic mapping study conducted by Jalali and Wohlin [13] into a systematic review. The previous study provided a classification and a visual summary of the type of research reports and results that have been published. The study methods and classification approaches in a systematic mapping and a systematic review differ in terms of goals, breadth and depth [16]. Therefore, we have used both methods complementary. In this paper, the list of databases is expanded and the analysis is extended to research method, contribution, and the results of the included papers. Further, the discussions are enriched and detailed in order to better explain the current status of using Agile practices in GSE based on findings from the literature. Hence, the objective of this study is to first summarize the current research literature, and then to investigate which Agile practices have been used effectively in which GSE contexts.

The remainder of the paper is organized as follows. Section 3.2 gives a brief background and summarizes related work. Section 3.3 discusses the research methodology and explains different steps of conducting this systematic review. The results of the study are presented in Section 3.4, and discussion and observations around them are provided in Section 3.5. Finally, conclusions and future research directions are presented in Section 3.6.

3.2. Background and Related Work

The Agile practices and GSE alternatives are shortly presented in this section before putting Agile practices in the context of GSE. Moreover, related research work regarding Agile practices and GSE is summarized, and finally the motivations and objectives of this study are explained.

3.2.1. Agile Practices

Agile methods consist of a set of practices for software development that have been created by experienced practitioners [27], aiming at overcoming the limitations of plan-based approaches through considering changes of the system’s requirements [11]. Agility is defined as “flexibility” and “leanness” [6], and mentioned to be about “feedback and change” in a way to “embrace, rather than reject, higher rates of change” [23].

Agile approaches focus on establishing close collaboration between customers and developers, and delivering software within time and budget constraints. Since they rely on frequent informal face-to-face communication rather than providing lengthy documentation, the process is repetitive, adaptive, and minimally defined [3].

The key features of Agile methods are continuous requirements gathering, frequent face-to-face communication, Pair Programming, refactoring, continuous integration, early expert customer’s feedback and minimal documentation [4]. The most widely used methodologies based on the Agile principals are Extreme Programming (XP) and Scrum. However, other methods such as Feature Driven Development, Dynamic Systems Development Method, Crystal Clear method, and Lean development have been also used [1][10].

3.2.2. Global Software Engineering

Geographically distributed software development teams characterize distributed software development, whilst globally distributed teams characterize global software development [18]. In this study, we have considered both as GSE. The description of different terms related to GSE is inspired by [18], and the authors have only made minor changes and generalization presented as follows.

Outsourcing (Offshore/Onshore Outsourcing): An external company is responsible for providing software development services or products for the client company. When both subcontracting and client companies are located in the same country, it is known as onshore outsourcing.
Offshoring (Offshore Insourcing): A company creates its own software development centers located in different countries to handle the internal demand.

Distributed Team: Team members are spread in different locations and work remotely on different parts of the project (independent tasks) with or without any face-to-face interactions. The difference between a virtual and a distributed team is that virtual team members work jointly on the same tasks.

3.2.3. Agile Practices in Global Software Engineering

Although Agile methods are well suited when customers and developers are co-located and there is frequent interaction among them [3], several software organizations have reported their successful experience of incorporating Agile in distributed software development (e.g. [25][9]). However, there are challenges associated with this combination, and to get it to work effectively considerable effort is needed. The major difficulties are summarized as related to communication, personnel, culture, different time zones, trust, and knowledge management [4]. Nevertheless, various tactics and solutions are also reported by different software organizations to mitigate these challenges.

3.2.4. Related Work

Here, a summary of the previous relevant research is presented. Systematic review studies on Agile methods or global software engineering are briefly presented. In addition, studies that have partially explored the combination of any Agile method in any GSE context are introduced even though if they are not a systematic review study.

Dybå and Dingsøyr [10] conducted a systematic review of empirical studies of Agile software development up to 2005 resulted in identifying 36 relevant empirical studies. Besides the comprehensive analysis of the papers, the need to increase both the number and the quality of studies and to establish a common research agenda in the area of study is pinpointed.

In a systematic review study by Šmite et al. [21] the empirical evidence in GSE-related research literature has been investigated. The amount of empirical studies in the area was found to be relatively small, hence it is concluded that the GSE field is still immature. Hence, they have shed light on paths for future work for both researchers and practitioners.

Taylor et al. [22] conducted a study in 2006 to evaluate the usefulness of the existing research on Agile global software development for practitioners. The study included articles published between 2001 and 2005. They concluded that the published research is of minimal value to practitioners since they do not provide novel guidance particularly for distributed Agile. It is concluded that the current research of experience reports is similar to the guides available before introduction of Agile.

Bose [4] performed an interesting study in 2008. He selected 12 case studies from literature that claimed to be successful in distributed Agile software development, and summarized them. The cases were evaluated in comparison with the Agile manifesto to determine to what extent Agile values and principles are followed. He discovered some innovative reported solutions for overcoming the challenges of distributed Agile development. The conclusion was that although many solutions seemed to be unique for the context of the challenges, they can still suitably guide companies in establishing and running distributed Agile software development.

Paasivaara et al. [15] have described how Scrum practices were adopted to benefit from distributed software development. Multiple case studies were conducted and the collected lessons learned were summarized. In addition, they have summarized the results of literature review on practices used in distributed Agile software development. However, the main contribution is not to explore the previous work. Hence, a systematic literature review has not been conducted.

The only systematic literature review in the area is published in 2009, and is performed by Hossain et al. [12]. It reviews 20 primary papers and identifies challenges of using Scrum in global software development. Additionally, the best practices addressing the identified challenges have been extracted. The presented guidelines and conclusions can help both practitioners and researchers in the area.
3.2.5. Motivations and Objectives

Confirming the findings of the previous works [21][12], the existing research in the area is exploratory in nature and mostly reports the cases in which some challenges were faced and some strategies were applied. It is also confirmed that lessons learned in one context may not directly apply in another one [21]. Hence, a standard approach for applying Agile in GSE does not exist despite the existence of great interest in Agile methodologies from software industry.

Exploring previous research showed that a comprehensive systematic review that covers all Agile methods in all GSE settings does not yet exist. Such a systematic review helps identifying different conditions and factors, which affect the success of Agile methods in GSE contexts. Hence, this study aims at systematically reviewing and summarizing the existing research literature, and investigating which Agile practices have been used effectively in GSE contexts. The results and findings may help practitioners in visualizing the risks and benefits of Agile global software development, and hence improving the performance in their work. It also helps researchers in obtaining an overview of the status of the area and highlighting the gaps.

3.3. Research Method and Conduct

The research was designed to be a systematic literature review following the guidelines provided by Kitchenham and Charters [14]. The first phase of the study was to draw a systematic map, in which the guidelines on how to conduct a systematic review was considered along with guidelines provided for performing a systematic map by Petersen et al. [16]. This paper presents all steps taken in designing and conducting the systematic review, and the results.

3.3.1. Research Questions

Based on perceived need for conducting a systematic literature review in the area, the research questions for this study are as follows:

RQ1: What is reported in the peer-reviewed research literature about Agile practices in GSE?

In order to answer this question, the current research literature had to be explored and summarized through conducting a systematic literature review study.

RQ2: Which Agile practices in which GSE settings, under which circumstances have been successfully applied?

To answer this question, the results of the systematic review had to be synthesized comprehensively to identify the successful empirical cases reported in the literature and analyze them carefully.

3.3.2. Search Strategy

The research started with defining a suitable scope, which was initially set to cover all Agile practices in all types of distributed development. It led to setting the preliminary research questions, and identifying the keywords. The initial keywords were searched in well-known databases such as ACM Portal and IEEE Xplore. Based on the search results, the research scope, research questions, and keywords were refined, search strings were reformulated, and searches were re-conducted. Moreover, the list of databases was expanded to collect as many relevant papers as possible. In parallel, a list of key papers was generated, which was used as a validation list to ensure the reliability and relevancy of the searches and to evaluate the search strings. The summary of the process is shown in Figure 3.1.

3.3.3. Data Sources

In a progressive process as discussed previously, the following databases were used:

- ACM Portal (http://portal.acm.org)
- IEEE Xplore (http://ieeexplore.ieee.org)
- AIS (http://aisel.aisnet.org)
- Inspec (http://www.engineeringvillage2.org)
- Compendex (http://www.engineeringvillage2.org)
- Scopus (http://www.scopus.com)
3.3.4. Data Retrieval

Search strings were formulated by combining different Agile practices and different types of distribution. It can be summarized as: (X1 OR X2 ... OR Xn) AND (Y1 OR Y2 ... OR Yn), where X covers most common Agile practices and Y includes different alternatives of GSE as presented in the following.

![Figure 3.1. Search strategy and process](image)

**X**: \{Agile, Scrum, Extreme Programming, Pair Programming, lean development, lean software development\}

**Y**: \{global software engineering, global software development, distributed software engineering, distributed software development, GSE, GSD, distributed team, global team, dispersed team, spread team, virtual team, offshore, outsource, open source\}

Agile practices were limited to Scrum, Extreme Programming, Pair Programming, and lean software development, intending to cover the most common ones, which are mostly used in practice. In addition, the objective was to ensure a clear focus on the scope of the systematic review. However, all spelling alternatives of keywords were considered (e.g. offshore, offshoring, off-shore, offshored, etc).

Furthermore, some limitations were applied on the searches. The written language was set to be English and the publication year was set to be between 1999 and 2009 with the purpose of summarizing the updated relevant related work in approximately the past decade.

In order to reduce the number of irrelevant hits, the search places were limited to title, abstract, and keywords. It should be noted that only peer-reviewed publications were taken into consideration and gray literature has not been explored.

3.3.5. Inclusion Process

The steps taken to extract the final set of studies for further synthesis are summarized in Figure 3.2. The searches resulted in identifying 534 papers. The decision on inclusion/exclusion criteria was made based only on the abstract due to two reasons. First, it is infeasible to evaluate the full text of 534 papers, and second full-text was not available for all papers. Based on the evidence found in the title, abstract or keywords implicitly or explicitly, the papers were categorized as “relevant”, “irrelevant” or “maybe relevant”.

Although the search strategy was carefully planned in a way to minimize the number of irrelevant or out of scope papers in the result of searches, many papers were judged as out of scope (e.g. not fit into the software engineering discipline). Hence, we put them in the irrelevant category as well.
In order to decrease the single researcher’s bias at this stage, the list of “irrelevant” and “maybe relevant” ones was given to the second researcher without showing the previous judgments. The result of the second judgment was slightly different regarding the “irrelevant” papers. However, it was decided not to include the papers with one “irrelevant” vote and one “maybe relevant”. Papers that both researchers classified as “maybe relevant” were included in the further analysis.

Finally, both researchers agreed upon a final set of papers for in-depth analysis. If the full paper was not accessible, an email sent to the main or second author asking for the paper in pdf. At the analysis step of this study, two emails remained unanswered, so those two papers were excluded. In addition, papers with no result or the same content as other studies were excluded. Thus, 81 studies were finally selected as primary papers for data extraction and synthesis.

3.3.6. **Data Extraction and Synthesis**

The guidelines provided by Petersen et al. [16] were used to build the classification scheme. Although they have suggested exploring the text adaptively if the abstract was not well structured, we decided to study full-text. We piloted a few studies and realized that critical information such as Agile practices, distribution type, and research method could not be extracted only from the abstract.

MS Excel was used for data extraction and collection (see Appendix 3.2). The items in the form were selected in alignment with the objectives of this study aiming at enabling the authors to answer the research questions by analyzing the extracted data.

The classification scheme suggested by Wieringa et al. [24] was used as a basis for determining the research type for the set of papers. A short description of each category, which was considered in this study, is provided below.

- **Evaluation Research**: Techniques or solutions are implemented and evaluated in practice, and the consequences are investigated.
- **Validation Research**: Techniques are novel, but still have not been implemented in practice. This is typically a study of a technique in a laboratory environment.
- **Solution Proposal**: A solution for a problem is proposed, and the benefits are discussed. The difference between a solution proposal and a validation research is in the level of abstraction for suggested solutions, which is higher for solution proposals.
- **Philosophical Paper**: It structures the area in the form of a taxonomy or conceptual framework, hence sketches a new way of looking at existing things.
Experience paper: It includes the personal experience of the author on what and how something happened in practice.

Opinion Paper: The personal opinion on a special matter is reflected in an opinion paper without relying on related work and research methodologies.

All 93 papers were fully read and 12 were excluded at this stage because either the results were not reported or the same study was reported more than once. Hence, data analysis was made for 81 remaining papers, and the required items were extracted, coded, and stored in Excel. Finally, several descriptive classifications of the content of the studied papers were made with respect to research methodology, empirical background, findings, participants, and context of the studies.

3.4. Results

The data required for analysis was extracted by exploring the full-text of each included paper. This section presents the collected data.

3.4.1. Results of Literature Review

The outcome of the selection phase was 81 peer-reviewed papers and articles. Table 3.1 shows the distinctive number of papers for each year (1999-2009). The maximum was in 2008 with 20 papers, and no relevant paper was found in 1999, 2000, and 2001 as well as few papers in 2002 and 2003. This indicates that GSE and Agile in combination has received more attention in the last five years. This is not surprising given that the interest for both Agile and GSE have increased during the last 5-10 years.

<table>
<thead>
<tr>
<th>Year</th>
<th>ACM</th>
<th>IEEE</th>
<th>Compendex</th>
<th>Inspec</th>
<th>AIS</th>
<th>Scopus</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2001</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2002</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2003</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2004</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2005</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2006</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2007</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2008</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2009</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The classification scheme explained in Section 3.3.6 was used for classifying the papers based on the research type. The results of the categorization are presented in Figure 3.3. It shows that the majority of the current literature is in the form of experience reports, in which practitioners have reported their own experiences on a particular issue and the method used to mitigate it. The distribution of different research types over studied years pinpoints the need for conducting more philosophical, validation, and evaluation research. Although experience reports are valuable, evaluation and validation research with a rigorous research method is required to establish foundations for a more mature area.

Furthermore, the collected data was processed to check which Agile practices had been applied in which distribution settings (see Figure 3.4). The current literature is mostly using “Agile” as a general term, and the term “distributed team” seems to be the most used team/organization setting in GSE. However, 12 studies did not report the context, and it was not derivable from the full-text of their studies. The lack of context and the quite general formulations regarding Agile and team make it difficult for others to make use of the findings [20].

More details and elaborations on the current available research are given in the following section when the successful cases are considered for further analysis. We excluded the failure stories in alignment with the research questions.
3.4.2. Successful Applications

Among all included papers, 63 empirical studies were found. Practitioners have written 40 papers, and academic researchers 20 of them. Three papers were found that have been written jointly by practitioners and academia. In total, 53 success stories were reported in the literature. If a report discussed N projects, the success/failure number for each of them was counted as 1/N. For example, if a paper reports two projects, one successful and one failure, we have added 0.5 to the successful cases and 0.5 to the failure ones.

Figure 3.3. Distribution of research types over the studied years

Figure 3.4. Mapping Agile practices and distribution types
The most used combination of Agile methods and distribution settings are Agile-offshore, Extreme Programming (XP)-distributed teams, and Agile-distributed teams. In the majority of the studies papers, the applied Agile method is addressed as “Agile” and distribution setting is mentioned as “distribution team” without any detailed information. It indicates the incompleteness of the contextual and background information in the current literature. Although XP is reported in many papers, too few practices were documented and enough information on this regard was not provided.

### 3.4.2.1. Countries Involved in Agile GSE

The countries involved in Agile GSE are summarized in Table 3.2. Countries represented as customer are: the main sites or the offices with major responsibilities in offshore developments, or the customers in outsourcing business relationships. If N countries are involved in a single relationship, the participation number for each was considered as 1/N. If more than one project was reported, the number was also divided by the number of projects.

<table>
<thead>
<tr>
<th>Countries involved in Agile GSE</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>10.5</td>
</tr>
<tr>
<td>Finland</td>
<td>1</td>
</tr>
<tr>
<td>Australia</td>
<td>1</td>
</tr>
<tr>
<td>Norway</td>
<td>1</td>
</tr>
<tr>
<td>China</td>
<td>1</td>
</tr>
<tr>
<td>Canada</td>
<td>0.5</td>
</tr>
<tr>
<td>France</td>
<td>0.5</td>
</tr>
<tr>
<td>Italy</td>
<td>0.5</td>
</tr>
<tr>
<td>Brazil</td>
<td>0.5</td>
</tr>
<tr>
<td>Czech Republic</td>
<td>0.5</td>
</tr>
<tr>
<td>Russia</td>
<td>0.5</td>
</tr>
<tr>
<td>Israel</td>
<td>0.5</td>
</tr>
<tr>
<td>Norway</td>
<td>0.5</td>
</tr>
<tr>
<td>Finland</td>
<td>1</td>
</tr>
<tr>
<td>Latvia</td>
<td>0.3</td>
</tr>
<tr>
<td>Malaysia</td>
<td>1</td>
</tr>
<tr>
<td>Hong Kong</td>
<td>0.3</td>
</tr>
<tr>
<td>Unclear</td>
<td>3.2</td>
</tr>
<tr>
<td>Supplier</td>
<td></td>
</tr>
<tr>
<td>India</td>
<td>10.5</td>
</tr>
<tr>
<td>USA</td>
<td>3</td>
</tr>
<tr>
<td>Ireland</td>
<td>2</td>
</tr>
<tr>
<td>Philippines</td>
<td>1</td>
</tr>
<tr>
<td>Ukraine</td>
<td>1</td>
</tr>
<tr>
<td>Poland</td>
<td>1</td>
</tr>
<tr>
<td>China</td>
<td>1</td>
</tr>
<tr>
<td>Canada</td>
<td>1</td>
</tr>
<tr>
<td>France</td>
<td>1</td>
</tr>
<tr>
<td>Italy</td>
<td>1</td>
</tr>
<tr>
<td>Brazil</td>
<td>1</td>
</tr>
<tr>
<td>Czech Republic</td>
<td>1</td>
</tr>
<tr>
<td>Russia</td>
<td>1</td>
</tr>
<tr>
<td>Israel</td>
<td>1</td>
</tr>
<tr>
<td>Norway</td>
<td>1</td>
</tr>
<tr>
<td>Finland</td>
<td>1</td>
</tr>
<tr>
<td>Latvia</td>
<td>0.3</td>
</tr>
<tr>
<td>Malaysia</td>
<td>1</td>
</tr>
<tr>
<td>Hong Kong</td>
<td>0.3</td>
</tr>
<tr>
<td>Unclear</td>
<td>3.2</td>
</tr>
</tbody>
</table>

Table 3.2. Countries involved in GSE
The collaborations between USA and India are reported the most in literature. According to Table 3.2, distributed development within USA is also popular. There are no Asian countries among the customers while some Asian countries are popular destinations for outsourcing such as India and Malaysia. The main reason could be due to availability of low-cost workforce.

3.4.2.2. Research Methods

As Figure 3.3 revealed, the majority of the current research is in the form of experience report. This fact was confirmed when categorizing the papers based on their research method (see Figure 3.5). Most experience reports and opinion papers were categorized as qualitative or unclear, and the research method was identified to be either unclear or a case study. The terminologies and definitions are inspired by Creswell’s book [5] on research design approaches.

![Figure 3.5. Research Method classifications for the studied papers](image)

As Figure 3.5 shows, 88% of the successful cases were qualitative studies in which either a case study was reported or analyzed. Only 2% of the cases were quantitative in which an experiment was conducted, and comparisons or evaluations were made, and 4% of them used both qualitative and quantitative approaches. However, the research method could not be identified for 6% of the studies.

Once more it highlights the need for conducting more of other types of research in this area. A large number of “Unclear” type of research methodology also indicates that more documentation on research design and conduct is required.

![Figure 3.6. Contribution and means of analysis of the papers](image)
3.4.2.3. Contributions and Means of Analysis

Figure 3.6 presents the contributions and means of analysis for the studied papers. As it was expected from previous analyses on research types and methods, the majority (70%) represents problem reports and lessons learned as their contribution. Some studies (11%) present recommendations when working Agile in a global context, and some authors presented the best practices applied in their organizations (6%). Very few papers (4%) explored industrial case studies and presented their results of analysis, which was basically the issues and solutions involved in Agile GSE. 4% of the cases developed tools to help Agile distributed development, and the rest (4%) was focused on comparisons between performance of Agile and co-located development.

3.4.2.4. Details of Successful Cases

All the primary studies were investigated to find out which Agile method was combined with which distribution setting, which practices were successfully applied for that combination and what countries were involved. In addition, the main characteristics of the project such as size, domain, and duration were extracted (see Appendix 3.3 and Appendix 3.4 for details). In the following, a brief summary is presented for each combination of Agile method and distribution according to the reported information in the studied papers.

In most cases, the team was distributed around the globe, working for a long time period on a small to medium size project. This can be concluded from the following sections. The project size was judged based on the following assumption: Small <= 20-person < Medium <= 50-person < Large. The duration was considered short if it was less than one month, and long if longer than 7 months. The specification of knowledge areas is based on SWEBOK [2].

The following Agile practices are extracted from the studied papers using their wordings. Some assumptions have made by authors that are listed in Appendix 3.3.

**Extreme Programming – Offshore:** In XP-Offshore combination, USA-India collaboration seemed to be the most popular, and “Retrospectives” is reported as the most efficient practice.

**Extreme Programming – Outsource:** Here, “Continuous Integration”, “unit/integration testing”, and “simple design” were practiced the most. USA was the owner of most projects and it was outsourced to China or within USA.

**Extreme Programming – Distributed team:** Using “Scrum/iterations” and “standup meetings” was the most effective practice in XP-distributed team setting. However, due to insufficient details provided in the related papers, we could hardly figure out how the tasks were accomplished and distributed among remote sites. However, USA seemed to be the owner of most projects.

**Extreme Programming – Virtual team:** Only one paper was found that addressed XP and Virtual team, and too few practices such as “standup meetings”, “automated testing”, “Pair Programming”, “onsite/proxy customer”, and “enough documentation” were reported. However, countries involved were not clearly specified.

**Scrum – Offshore:** Many papers were found that addressed this type of collaboration. The initiator of offshoring was USA in most cases and India or China was chosen as destination. The most reported Scrum practices are “Sprint/iterations”, “Retrospectives”, and “Sprint review/demo”.

**Scrum – Outsource:** The outsourcing company was mostly located in USA, and “Pair Programming”, “one team/sit together”, “Scrum of Scrum”, and “continuous/automated builds” were reported as the most successful practices.

**Scrum – Distributed team:** USA is the country, which is involved the most, and the most efficient practices are “standup meetings” and “backlog”.

**Agile – Offshore:** In this case, the initiator of offshoring was also mostly located in USA. The efficient Agile practice is reported to be “Sprint planning”.

**Agile – Outsource:** “Continuous Integration” was reported the most in this combination, and Denmark and Russia were the most popular countries.

**Agile – Distributed team:** In this setting, India was the most popular country, and “standup meetings” and “sprint/iterations” were the most popular practices.
Agile – Open source: “standup meetings”, “Pair Programming”, “sprint/iterations”, “Test Driven Development”, and “unit/integration testing” were the most efficient Agile practices in this combination, and Italy and Norway seemed to be the more involved countries.

3.4.2.5. Successful Agile Practices

All empirical studies that reported successful cases were explored in order to identify the applied Agile practices. The practices and their frequencies are summarized in Figure 3.7. According to the available research literature, the frequencies represent that the “standup Scrum meetings”, “sprint/iterations”, “continuous integration”, and “sprint planning” are the activities, which are efficiently practiced the most. The frequency of 18.5 for “standup Scrum meetings” indicates that 18.5 cases out of 53 reported successful application of this practice.

Although several practices were reported in the literature, in many cases it was unclear which Agile method has been particularly used. It was also observed that some cases claimed to be Agile while too few practices were actually used. Hence, the reliability of their findings cannot be ensured. As a consequence, extra caution is required when using their best practices.

![Figure 3.7. Agile practices and their frequencies in the studied papers](image)

3.4.2.6. Efficient Agile Method - Distribution Type Combination

Extreme programming in combination with globally distributed team (9 identical papers) has been reported as the most efficient Agile-GSE setting in the current literature. Then, Agile in combination with offshore (7.5 identical research papers) is reported the most. The list of all identified combinations in the literature is presented as follows sorted by their popularity.

1) XP – Distributed team: 9
2) Agile – Offshore: 7.5
3) Scrum – Distributed team: 7
4) Scrum – Offshore: 6.5
5) Agile – Distributed team: 6
6) Scrum – Outsource: 4.5
7) XP – Offshore: 3
8) XP – Outsource: 3
9) Agile – Open source: 2
10) Agile – Outsource: 1.5
11) Agile – Virtual team: 1
12) XP – Unclear: 1
13) XP – Virtual team: 1
14) PP – Distributed team: 1
However, in most cases in which “distributed team” was presented as the distribution type, further information was not provided. Hence, it was difficult to extract the exact form of collaboration or task distribution among remote sites. The same discussion is valid for the cases in which “Agile” was reported as applied Agile method.

3.4.3. **Limitation**

The major concern with any type of research is the reliability. Therefore, two researchers were involved in this systematic review study, discussing the reliability threats early in the design phase. The procedure was discussed and agreed considering the activities to mitigate the effect of one researcher’s bias.

The results of the searches were judged for inclusion/exclusion jointly as discussed in Section 3.3.5. The co-researcher reviewed one random paper, which was previously reviewed by the leading researcher of this study. The purpose was to measure the differences between the results of their data extraction, aiming at minimizing the bias and increasing the accuracy in data collection and categorization.

In order to address the conclusion validity, we collected as many papers as possible from a variety of sources including ACM, IEEE, AIS, Inspec, Compendex, and Scopus online catalogues. Although different disciplines use different terminologies (e.g. for distributed team), we included as many alternatives as possible for the keywords when formulating the search strings. In addition, the publication year was set to be from 1999 to 2009, which was wide enough to capture most of the relevant publications due to the fact that common Agile practices are not much older than one decade. So, it was possible to observe the trends in the area over the past decade.

However, replicating this study may result in a slightly different set of papers, both in searching in the databases and in inclusion/exclusion process.

We kept the gap between conducting searches in different sources less than one week, and finally updated the results in January 2010 to ensure capturing all studies published in 2009 (or at least entered into the databases before the end of 2009).

Some papers may have been missed due to application of constraints on the search strings in order to reduce the number of irrelevant papers found in the searches. We do not claim to have collected all relevant studies, but we included as many studies as possible. It should also be noted that although some studies may have been missed, there is no reason to believe that they would be distributed differently across the classifications than the papers included in the systematic review presented.

Since many empirical papers that we studied did not provide sufficient contextual details, we derived some data from the text (e.g. project size and duration). It has been impossible to judge the reported content separately, hence we trusted authors about what they reported on Agile practices, distribution type, and the success of the project. It may have led to some unwanted inaccuracies in data extraction process. Furthermore, inconsistencies in reporting contextual information (e.g. in documenting the level of details) in the studied research literature may also have caused some inaccuracy in our data analysis. For example, some studies reported several practices whilst it was unclear which Agile method has been particularly used. In the other hand, some cases claimed to be Agile while too few practices were used. Therefore, extra caution is needed when applying their best practices in other situations.

In responding to the research question 2, we have analyzed only successful empirical cases. Although failure stories may be useful as well, we did not include them because we wanted to check only the success reports. However, the number of failure cases was not large to influence the results and conclusions of this study dramatically.

In summary, we can claim that although the findings of similar studies may be slightly different from the findings of this research regarding numbers and figures, it will not change the patterns we have identified in the results.

3.5. **Discussions**

The following sections present some discussions based on our investigations and observations on the results of this systematic review study.
Growing Interest: The applicability of Agile practices in GSE is not yet well investigated. It is clear that several challenges are associated with combining them. However, an increasing number of publications, in particular experience reports, in last five years, indicates a growing interest in this area from software industry.

Based on the explored papers, we cannot conclude that globally distributed software development is becoming more popular in software organizations comparing to Agile software development and vice versa. In some cases, an Agile organization decided to expand its offices [T56], and in some others, a distributed company decided to switch to Agile e.g. due to failure of a process-driven development approach [T22]. Hence, we can only conclude that Agile and GSE in combination has attracted more attention in past five years.

Research Type: The majority of the existing research literature is in the form of industrial experience reports. It reveals the need for conducting more evaluation research by which actual practices will be comprehensively examined. This type of research requires rigorous research methods and literature reviews, so one possible option could be close collaboration of industry and academia in this area. The research part can be done in academia while data has to be collected from real industrial cases. Further, the characteristics of academic environments are different form industry e.g. in market and business aspects and interaction with customers. Therefore, it is very challenging to run industrial projects in pure academic environments.

Repetitions: We observed some repetitions in the content of the studies we explored. Similar problems are reported more than once in different articles (e.g. [T74][T22]). It may indicate that previous research is not studied in software organizations or it is hard to interpret the context of different experiences. Another evidence for this conclusion is that industrial experience reports do not normally include the related work and do not reference to the literature. However, it requires further investigation to realize whether the academic materials such as textbooks or research papers are of interest for industry in this specific area.

Another type of repetition is when the same problem is reported, but different solutions are proposed (e.g. [T24][T71]). It might be due to the several differences between different organizations, nations, and projects. It may mean that solutions are dependent on the situational needs of the organization or project as well as the people involved in decision-making. This fact threatens the generalizability of many studies found in this systematic review since they were experiences from a single case study with insufficient information on the context. Hence, the practical applicability of these studies can be doubted, so further analyses and evaluations can be made on the current literature to ensure its usefulness for any future research.

Corresponding Challenges: There are not a sufficient number of studies analyzing the challenges of applying Agile in GSE. Problems and challenges are documented in GSE or Agile, while the combination is not well examined in real world situations. Some academic studies suggested that Agile mitigates GSE challenges ([T42][T16]), whilst others believe they are contradictory in nature and it emphasizes the GSE challenges [T9]. Hence, we conclude that there is a need for in-depth studying of challenges and benefits of combining Agile and GSE in the form of evaluation research.

We also suggest creating and maintaining a universal database (e.g. an online library), which contains and maintains several reported challenges and various solutions to each, along with keeping record of situation specific information. This database can be open-sourced updated directly by practitioners.

Contextual Information: As mentioned previously, the contextual detail for many empirical studies in this area is insufficient. Having this information assists researchers in examining the practical applicability of the reported cases for other settings. It demands researchers in this area to design and use a template for documenting the contextual information, which is not too detailed and not too abstract. We recommend practitioners and researchers to read guidelines presented by Petersen and Wohlin [17] and keep them in mind when writing their reports.

Scaling up Agile: According to the literature, Agile has been successfully applied in small to medium size distributed projects over a medium to long time period (Section 3.4.2.4). Therefore, there is not sufficient evidence to conclude that Agile is efficiently applicable in large distributed projects. Although few studies have reported their experiences of large projects such as [T40][T41], the other contextual project factors are not clearly reported.
Large Agile teams may face complications and issues in their communications [T31], and it may cause control/coordination problems for the project managers. Adding distance and distribution implies differences in time zones, cultures and working styles, which increases the complexity of communication, control, and coordination in Agile GSE. Although some studies e.g. [T31] proposed best practices and recommendations on how to deal with these issues, the guidelines generally address GSE-related issues rather than considering Agile and GSE in combination.

**Modified Agile Practices:** In many studies that we reviewed, Agile practices had been customized and a modified Agile method was applied [T77]. The motivations for these adjustments were reported to be distribution type, overlapping working hours or other factors depending on the situational requirements of the project.

Another type of modification that was observed was mixing different methods with selective set of practices from different methods e.g. [T24]. In some other cases, XP and Scrum practices were selectively applied in situations that is claimed to be either XP or Scrum e.g. [T64].

It highlights the need for further research in which the modifications are well studied in order to provide guidelines for practitioners on how to adapt the practices to their needs. In addition, the changes shall be compared to the original descriptions (e.g. Agile manifesto) and determine the safe variance of the changes to remain Agile, and of course efficient in software development. In other words, it shall be determined that how much change is allowed to be still recognized as practicing Agile in GSE.

### 3.6. Conclusions

The current research literature on the application of different Agile practices in GSE was summarized in this study. Further, the empirical studies that reported success cases were explored to investigate under which circumstances they have been efficiently practiced in software organizations.

Summarizing the relevant research literature provided the answer to the RQ1. The experience reports of working with globally distributed teams constitute the major part of the literature. They have contributed by explaining the issues, specific solutions, and the lessons learned. However, the majority of them have not documented the characteristics of their empirical study and the context under which the project was running.

The success reports were examined to find the answer to the RQ2. The existing literature mainly consists of successful empirical experiences in which globally distributed teams collaborate over a long time on small to medium sized projects (Section 3.4.2.4). XP combining with globally distributed team setting is reported the most in the efficient empirical cases in the current research literature (Section 3.4.2.6). USA-India is reported to be the most common applied distributed business relationship with successful results (Table 3.2).

Several practices were found in the literature, which have been applied in software organizations. The most common practices used according to the literature are “standup meetings”, “sprint/iterations”, “continuous integration”, “sprint planning”, “retrospectives”, “Pair Programming”, “sprint review/demo”, “Test Driven Development”, “Scrum of Scrum”, “onsite/proxy customer”, and “backlog”.

During the course of this study, we observed that practitioners and researchers have different perception of what exactly Agile practices are and how to report and document them. Therefore, there is a need for them to collaborate closely and illustrate the practices and agree on the terminology and how to document the context. It helps practitioners when setting up a new Agile GSE project in a way that they can find similar cases in the literature based on contextual/background information and check if similar practices are applicable in their projects as well.

In summary, the emergent need can be explained as developing a comprehensive framework that considers various factors from different perspectives when applying Agile in GSE. It can be used as a basis for decision-making in early phases of software development, and assists project managers in estimating the risks, challenges, and benefits of using Agile in (globally) distributed projects.

The results of this study will be used towards proposing such a comprehensive framework for Agile applicability in GSE. Currently, we are working on developing a model in order to provide a unified concrete basis for judgments about accordance to Agile values and principals in different organizational settings.
Acknowledgments

This work was partly funded by the Industrial Excellence Center EASE - Embedded Applications Software Engineering, (http://ease.cs.lth.se).

References


49

onshoring”, *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE)*, pp. 262-274.


Appendix 3.1. Included studies


2004 conference of the Centre for Advanced Studies on Collaborative research (CASCON ’04), IBM Press, pp. 242-254.


## Appendix 3.2. Data extraction form

### General

- **Publication year**: (1999-2009)
- **Database**: (ACM, IEEE, Inspec, Compendex, AIS)
- **Number of authors**: $\geq 1$
- **Authors’ background**: (industry, academic, unclear)
- **Affiliations**
- **Countries**

### Agile Practices

- **Main practice**: (Agile, Scrum, XP, Pair Programming, Lean)
- **Sub-practices**
- **Agility level**: (not all teams, all team, organization)

### GSE Settings

- **Distributed type**: (distributed team, virtual team, offshore, outsource, open source)
- **Global**: (yes, no, unclear)
- **Number of sites**: $\geq 1$
- **Countries**

### Research Methodology

- **Empirical**: (yes, no, unclear)
- **Research type**: (evaluation, validation, solution proposal, philosophical, personal experience, personal opinion)
- **Research method**: (qualitative, quantitative, mixed)
- **Research sub-method**: (single case study, multiple case study, experiment, literature review, etc)
- **Means of data collection**: (survey, questionnaire, interview, literature, etc)
- **Means of analysis**: (comparison, descriptive, measurement, classification, etc)

### Empirical Project Features

- **Size**: (small, medium, large, unclear)
- **Duration**: (short, medium, long, unclear)
- **Participants**: (industry, students, unclear)
- **Domain**: (telecom, oil industry, web based, real time, embedded, etc)
- **Knowledge area**: (requirements engineering, design, development, testing, tools, project management, quality, etc)
- **Successful**: (yes, no, unclear)

### Results

- **Contributions**: (problem report, recommendations, lessons learned, tools, framework, etc)
Appendix 3.3. Mapping practices and distributions

All the primary studies were investigated to find out which Agile method was combined with which distribution setting, which practices were successfully applied for that combination, and what countries were involved. In addition, the main characteristics of the project such as size, domain, and duration were extracted, and presented as follows. It should be noted that all presented data is extracted based on the provided contextual information in the studied papers e.g. for addressing the knowledge area. The numbers for each item represents its frequency i.e. how many times it is reported. The “Unclear” items means that it was not possible to clearly extract data from the text.

In extraction of practices, we have made some assumptions, which are:

- All synonyms are merged. For example, “customer in team” and “onsite customer” are considered the same.
- Practices for XP and Scrum are not merged even thought they are basically the same and only with different wordings. For example, “Scrum planning” is not merged with “Planning game”.
- The names of practices are extracted and borrowed from the studied papers although it is not completely matched with the Agile references.
- Practices with frequency below two are not presented.
- “Standup calls” is considered equal to “standup meetings”, “standup Scrum meetings”, and “distributed standup meetings”.
- If a paper claimed that all practices were applied and the reference was not provided, we excluded that paper because we have not set a reference to check all reported practices with, and instead, have reflected the reported Agile practices.

A. XP – Offshore [T2][T14][T40][T58]

<table>
<thead>
<tr>
<th>Countries:</th>
<th>USA: 1.5, India: 1, Philippine: 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project domain:</td>
<td>Automotive: 1, Unclear: 3</td>
</tr>
<tr>
<td>Project duration:</td>
<td>Medium: 1, Unclear: 3</td>
</tr>
<tr>
<td>Project size:</td>
<td>Large: 1, Unclear: 3</td>
</tr>
<tr>
<td>Knowledge area:</td>
<td>Construction: 2, Testing: 2, Design: 1, Maintenance: 1, Unclear: 2</td>
</tr>
</tbody>
</table>

B. XP – Outsource [T41][T38][T75]

<table>
<thead>
<tr>
<th>Countries:</th>
<th>USA: 1.5, China: 0.5, New Zeeland: 0.5, Unclear: 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project domain:</td>
<td>Unclear: 3</td>
</tr>
<tr>
<td>Project duration:</td>
<td>Large: 2, Unclear: 1</td>
</tr>
<tr>
<td>Project size:</td>
<td>Large: 1, Medium: 1, Unclear: 1</td>
</tr>
<tr>
<td>Knowledge area:</td>
<td>Requirement: 1, Design: 2, Construction: 2, Testing: 1, Unclear: 2</td>
</tr>
</tbody>
</table>

C. XP – Distributed team [T19][T22][T24][T31][T35][T47][T69][T74][T78][T79][T81]

<table>
<thead>
<tr>
<th>Countries:</th>
<th>USA: 3.8, Ireland: 1, Poland: 0.5, Finland: 1, Czech Republic: 0.5, Brazil: 0.5, India: 0.8, Australia: 0.5, Malaysia: 0.5, UK: 0.3, Unclear: 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project domain:</td>
<td>Real-time: 1, Telecom: 1, Commercial: 1, Web: 3, Service: 1, Unclear: 3</td>
</tr>
<tr>
<td>Project duration:</td>
<td>Long: 2, Medium: 1, Unclear: 8</td>
</tr>
<tr>
<td>Project size:</td>
<td>Small: 2, Unclear: 9</td>
</tr>
<tr>
<td>Knowledge area:</td>
<td>Testing: 1, Construction: 3, Design: 1, Requirement: 1, SE Management: 2, Unclear: 6</td>
</tr>
</tbody>
</table>

D. XP – Virtual team [T56]

<table>
<thead>
<tr>
<th>Countries:</th>
<th>Unclear: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project domain:</td>
<td>Unclear: 1</td>
</tr>
<tr>
<td>Project duration:</td>
<td>Long: 1</td>
</tr>
<tr>
<td>Project size:</td>
<td>Short: 1</td>
</tr>
<tr>
<td>Knowledge area:</td>
<td>Requirement: 1, Design: 1, Construction: 1, Testing: 1</td>
</tr>
<tr>
<td>Knowledge area</td>
<td>Project size</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>E. Scrum – Offshore [T13][T14][T33][T44][T57][T58][T70][T71]</td>
<td>Large: 1, Small: 3, Unclear: 4</td>
</tr>
<tr>
<td>F. Scrum – Outsource [T18][T33][T62][T63][T64]</td>
<td>Medium: 1, Large: 1, Small: 2, Unclear: 5</td>
</tr>
<tr>
<td>G. Scrum – Distributed team [T5][T15][T24][T43][T59][T77][T78][T79][T81]</td>
<td>Medium: 1, Large: 1, Small: 2, Unclear: 5</td>
</tr>
<tr>
<td>H. Agile – Offshore [T3][T10][T12][T30][T32][T39][T60][T66]</td>
<td>Large: 1, Medium: 2, Small: 1, Unclear: 4</td>
</tr>
<tr>
<td>I. Agile – Outsource [T60][T45]</td>
<td>Small: 1, Medium: 1</td>
</tr>
<tr>
<td>J. Agile – Distributed team [T17][T29][T51][T49][T76][T80]</td>
<td>Medium: 2, Large: 1, Small: 1, Unclear: 2</td>
</tr>
<tr>
<td>K. Agile – Virtual team [T20]</td>
<td>Unclear: 1</td>
</tr>
<tr>
<td>L. Agile – Open source [T34][T68]</td>
<td>Large: 1, Unclear: 1</td>
</tr>
</tbody>
</table>
M. Pair Programming – Distributed team [T55]

<table>
<thead>
<tr>
<th>Category</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Countries</td>
<td>Unclear: 1</td>
</tr>
<tr>
<td>Project domain</td>
<td>Education: 1</td>
</tr>
<tr>
<td>Project duration</td>
<td>Unclear: 1</td>
</tr>
<tr>
<td>Project size</td>
<td>Unclear: 1</td>
</tr>
<tr>
<td>Knowledge area</td>
<td>Tools &amp; Methods: 1</td>
</tr>
</tbody>
</table>
Appendix 3.4. Context elements for each Agile practice

For each extracted practice, we have visualized the information regarding in which specific setting it has been applied. It is presented in Figure 1 to Figure 6.

Figure 1. Context for “standup meetings”, “sprint/iterations”, “continuous integration”, “sprint planning”
Figure 2. Context for “retrospectives”, “Pair Programming”, “sprint review/demo”, “test driven development”

Figure 3. Context for “Scrum of Scrum”, “Onsite/proxy customer”, “backlog”, “unit/integrated testing”
Figure 4. Context for “coding standards”, “refactoring”, “planning game”, “continuous/automated builds”

Figure 5. Context for “automated testing”, “collective code ownership”, “simple/incremental design”, “user stories”
Figure 6. Context for “architecture focus”, “burn down charts”, “one team/sit together”, “enough documentation”
Chapter 4

Systematic Literature Studies: Database Searches vs. Backward Snowballing

Abstract

Systematic studies of the literature can be done in different ways. In particular, different guidelines propose different first steps in their recommendations, e.g. start with search strings in different databases or start with the reference lists of a starting set of papers. In software engineering, the main recommended first step is using search strings in a number of databases, while in information systems, snowballing has been recommended as the first step. This paper compares the two different search approaches for conducting literature review studies. The comparison is conducted by searching for articles addressing “Agile practices in global software engineering”. The focus of the paper is on evaluating the two different search approaches. Despite the differences in the included papers, the conclusions and the patterns found in both studies are quite similar. The strengths and weaknesses of each fits step are discussed separately and in comparison with each other. It is concluded that none of the first steps is outperforming the other, and hence the choice of guideline to follow, and hence the first step, may be context-specific, i.e. depending on the area of study.

Keywords


4.1. Introduction

Research literature may be divided into primary studies (new studies on a specific topic) or secondary studies (summarizing or synthesizing the current state of research on a specific topic). The secondary studies may be used to pinpoint gaps or to highlight areas that require more attention from researchers or practitioners.

Secondary studies require comprehensive searches in the published research literature. Kitchenham and Charters [5] proposed a systematic literature review (SLR) approach inspired from evidence-based medicine, which recommend starting with systematic searches in databases using well-defined search strings to find relevant literature. In the guidelines [5], it is recommended that snowballing from reference lists of the identified articles should be used in addition to the searches in the databases, i.e. to identify additional relevant articles through the reference lists of the articles found using the search strings.

However, the guidelines do not explicitly recommend forward snowballing, i.e. identifying articles that have cited the articles found in the search and backward snowballing (from the reference lists). In our experience, most systematic literature reviews (including our own) do not use snowballing as a complement to searching the databases. It is fully understandable given the amount of work needed to conduct a systematic literature review. The implication being that a review provides a limited set of all papers on the topic, i.e. a sample of the population.
Webster and Watson [2] proposed a slightly different approach to systematic literature studies in the field of information systems. They propose to use snowballing as the main method to find relevant literature. In their recommendation, they highlight both backward snowballing (from the reference lists) and forward snowballing (finding citations to the papers). The snowballing approach requires a starting set of papers, which they suggest should be based on identifying a set of papers from leading journals in the area.

Given that there exist different guidelines of how to conduct a systematic literature study, we pose the following research questions (RQ):

RQ1: To what extent do we find the same research papers using two different review approaches?

RQ2: To what extent do we come to the same conclusions using two different review approaches?

The outcome of a systematic literature study is either a systematic literature review [5] or a systematic mapping study [13]. Database searches and snowballing are by no means the only options. The use of personal knowledge/contacts [6], or mixed methods [11] has also been discussed in the literature. The focus here is, however, on the first step of recommended method to identify relevant literature, since it is of course always possible to combine all different ways of finding literature. Thus, we have limited our study to using either database searches as a first step or backward snowballing as a first step, in particular given that, in our experience, researchers are, all to often, forced to limit their search procedures given the time it takes to conduct a systematic literature study. The papers found are evaluated for relevance and quality, which gives a set of primary studies for each search approach (database search or backward snowballing).

Given that the number of published secondary studies increases [3], it is perceived as important to understand whether or not the first step in the searches impacts the actual outcomes of the systematic literature study, in particular since many published papers do not use all steps recommended in the guidelines. This is closely related to the need to ensure reliability of secondary studies, which means whether two independent studies on the same topic would find the same set of papers and draw the same conclusions [9].

Based on the need identified, we conducted two different literature reviews on Agile practices in Global Software Engineering (GSE) using different guidelines for the literature search, and in particular we only used the first step in the recommendation, i.e. database searches [5] or backward snowballing [2]. It should be noted that we did include distributed development within a country in GSE too. The main reason being that many of the challenges experienced in a global setting also occurs in distributed development within a country, although some of the challenges are amplified when going global. Both studies have the same research questions. The first study is an SLR (presented in Chapter 3), and the second study applied a snowballing approach [2]. The differences between the search methods are discussed in more detail in Section 4.3.

The remainder of the paper is structured as follows. Section 4.2 summarizes related work, and Section 4.3 discusses the research method and introduces the two studies forming the input to the analysis. The results are presented in Section 4.4, and the discussions on findings are given in Section 4.5. Finally, Section 4.6 presents the conclusions and the future research directions.

4.2. Related Work

Inspired from medicine, in which systematic literature reviews is an approach for synthesizing evidence, Kitchenham et al. [1] introduced the concept of evidence-based software engineering (EBSE). A couple of years earlier Webster and Watson [2] suggested a structured approach in information systems to conduct systematic literature studies.

It should be noted that the research type in medicine and software engineering (SE) are not necessarily the same (e.g. controlled experiments vs. case studies). It implies different types of data, and different types of analyses on the data. Hence, the synthesis of the data collected from an SLR in SE may not be as straightforward as in at least some parts of medicine.

However, a practitioner-oriented view was formulated based on the EBSE ideas [4], and researchers also suggested guidelines for conducting systematic literature reviews [5]. Furthermore, Brereton et al. [7] reviewed a number of existing literature reviews to examine the applicability of SLR practices to SE. They found out that although the basic steps in the SLR process are as relevant in SE as in medicine, some modifications are necessary for example in reporting of empirical studies in SE.
Although the number of literature review studies in SE has increased in the past five years [3], few studies exist which evaluate the reliability of literature search approaches for example to evaluate the repeatability of protocol-driven methods or to compare the results of literature searches conducted through different methods such as SLR and snowballing. In the following, we summarize the relevant research.

Greenhalgh and Peacock [6] conducted a study in order to describe where papers come from in a systematic review of complex evidence. They applied three different methods and found 495 primary sources related to “therapeutic interventions”. Their conclusion was that protocol-driven search strategies by themselves are not the most efficient method regardless of the number of traversed databases, because some sources may be found through personal knowledge/contacts (e.g. browsing library shelves, asking colleagues), and snowballing is the best approach for identifying sources published in obscure journals.

In 2009, Skoglund and Runeson [8] investigated a reference-based search approach with the primary purpose of reducing the number of initial articles found in SLRs. Although the proposed method increased the precision without missing too many relevant papers for the technically focused reviews, its results were not satisfactory when the search area was wide or the searches included general terms. This implies that the choice of approach to searching is context-dependent.

Zhang et al. [10] conducted two participant-observer case studies to propose an effective way of identifying relevant papers in SLRs. The approach was based on the concept of quasi-gold standard for retrieving and identifying relevant studies, and it was concluded to serve the purpose and hence it can be used as a supplement to the guidelines for SLRs in EBSE. In a follow up validation study [11], a dual-case study was performed, and the proposed approach seemed to be more efficient than the EBSE process in capturing relevant studies and in saving reviewers’ time. Further, the authors recommended an integrated search strategy to avoid limitations of applying a manual strategy or an automated search strategy.

MacDonell et al. [9] evaluated the reliability of systematic reviews through comparing the results of two studies with a common research question performed by two independent groups of researchers. In their case, the SLR seemed to be robust to differences in process and people, and it produced stable outcomes.

Kitchenham et al. [12] conducted a participant-observer multi-case study to investigate the repeatability of SLRs performed independently by two novice researchers. However, they did not find any indication of repeatability of such studies that are run by novice researchers.

In summary, too few studies have addressed the reliability of secondary studies. As discussed here, they have either compared different SLRs or mapping studies to check whether the same results are achieved [9] and [12], or investigated more efficient approaches of searching [6], [8] and [10]. As a complement to previous studies, we investigate the reliability of secondary studies using different search strategies. This is done by comparing the outcome of two studies on the same topic using different guidelines for finding the relevant literature. The research method used is discussed next.

4.3. Research Method

The main objective of this study is to examine whether two systematic review studies would provide the same result when the applied first step in the search strategy is different. Therefore, we planned two separate literature reviews.

The first study was conducted within 2009-2010 to capture relevant research about the most common Agile practices applied in different settings of global software engineering (see Chapter 3). The second study was performed during 2010-2011 with exactly the same purpose and the same research questions. The difference between the two studies was the way that the relevant papers and articles were extracted from the published research, i.e. the search strategy.

The time between the two searches was a couple of months and the time between the syntheses was around eight months, and hence the details about specific papers found in the first search were not fresh in the mind of the researchers. Thus making the searches reasonably independent. An alternative would have been to have different researchers conducting the two studies.
However, this would have introduced threats in relation judgments of inclusion and exclusion of papers. The threat of having the same researchers involved was mitigated by time, i.e. by leaving several months between the two studies the researchers did not remember all the details of individual papers.

The first study (study 1) follows the guidelines provided by Kitchenham and Charters [5] as far as it comes to conducting searches in the databases. Study 1 did not use snowballing from reference lists as recommended in the guidelines. In the second study, a backward snowballing [2] approach was used. The starting set for the snowballing approach was generated through a search in Google Scholar\(^1\). This is further elaborated below.

The snowballing search method [2] can be summarized in three steps: 1) Start the searches in the leading journals and/or the conference proceedings to get a starting set of papers, 2) Go backward by reviewing the reference lists of the relevant articles found in step 1 and step 2 (iterate until no new papers are identified), and 3) Go forward by identifying articles citing the articles identified in the previous steps.

Based on that study 1 was focused on a specific time period, i.e. 1999-2009, it was decided to identify a starting set of papers from 2009 and then use backward snowballing based on the papers found. Given that researchers seem to focus on the database search, despite the guidelines [5], it was decided to only compare the first step for the searches, i.e. the database search vs. the backward snowballing approach. It was done for two reasons:

1. It would make the systematic literature review using the guidelines more representative of the state of studies actually published. As a consequence we saw a need to not follow the guidelines [2] for snowballing perfectly either. Thus, trying to be as fair as possible in the comparison. It would have been unfair to follow the guidelines very closely in one case and then not in the other case.

2. It was realized that a more comprehensive use of the guidelines, i.e. following all steps recommended, would result in the outcomes getting closer to each other. With the first step, we refer to only doing the database search. However, for the papers found in the database search, we check relevance and quality to finally have set of primary studies. The same procedure is done based on the other guideline [2], i.e. we only perform backward snowballing and identify a primary set of papers. Having done all steps recommended in the guidelines would undoubtedly mean that more papers would be included and if the searches being perfect they would end up with exactly the same set of papers. Thus, we wanted to compare the first steps in the different guidelines, since it is reasonable to believe that if these produce similar enough results than a larger sample of papers would just increase the similarity. Thus, we are concerned with comparing the samples of papers obtained when conducting the first steps in the two different guidelines [5] and [2] respectively.

Furthermore, to make the studies as comparable as possible, we kept the search terms and keywords as similar as possible in both studies and also applied the same constraints on searches. This means that the same search terms were used in the database searches in study 1 as in the Google Scholar search in study 2. In addition, the same researchers were responsible for finding, evaluating, and analysing the relevant papers in both studies in order to minimize the diversity in data collection and data analysis. Hence, the only (intended) difference between study 1 and study 2 is the search approach (the way we identified the relevant papers).

The assessment is performed through comparing the results of the two studies based on their primary papers and their conclusions. In summary, the research questions are:

RQ1: To what extent do we find the same research papers using two different review approaches?

RQ2: To what extent do we come to the same conclusions using two different review approaches?

In order to answer the research questions, we conducted an in-depth comparison of the two studies.

\(^1\) http://scholar.google.com/intl/en/scholar/about.html
4.3.1. Details of Studies

4.3.1.1. Study 1

It was designed to be a systematic literature review following the guidelines by Kitchenham and Charters [5], although only doing the database searches and not snowballing. The study was conducted during 2009-2010 with the purpose of capturing the status of combining agility with GSE. The results were limited to peer-reviewed conference papers and journal articles published in 1999-2009. The final set of papers (81 distinctive papers) was synthesized by classifying them into different categories (e.g. publication year, contribution type, research method and Agile practices used in GSE). More details of the study 1 can be found in Chapter 3.

4.3.1.2. Study 2

It had the same purpose and research questions as the first study, and was conducted after we were finished with study 1 (during 2010-2011) (presented in Chapter 6). In this study, we followed the guidelines provided by Webster and Watson [2] regarding identification of a starting set of papers followed by backward snowballing. We searched in Google Scholar using the same search strings as in study 1, and then limiting the search to 2009 to identify a starting set of papers. It should be noted that the search in Google Scholar was conducted only once, i.e. to identify the starting set of papers for the backward snowballing. First, we evaluated the relevancy of the papers and then went through the reference list of the relevant papers in order to find additional sources. The process was stopped when we could not add any further relevant papers published in the time period 1999-2009. The analysis of the data was kept as similar as possible to study 1. Some further details of study 2 can be found in Chapter 6, since our objective is not to present the individual studies as such, the focus is on comparing the outcome of the two different first steps for the searches based on guidelines by Kitchenham and Charters [5] and Webster and Watson [2] respectively. More details of the study 2 are presented in Chapter 6.

4.3.2. Comparison Approaches

The comparison is done in two different ways. First, we examined all papers identified in study 1 and study 2 regarding the papers included and the findings. However, due to the fact that the majority of the articles were identical, it was not surprising that the conclusions and the findings would be also similar in both studies. Therefore, we conducted a second comparison in which we excluded the papers, which were in common for both studies, and performed the analyses solely on the unique papers for study 1 and study 2 respectively. Then, we compared the findings from the two analyses.

4.4. Results

In the following, we present the differences and similarities between the findings of study 1 and study 2 given the different first steps for the searches.

4.4.1. Number of Papers

The first comparison relates to the number of papers found in the two studies. Study 1 resulted in 534 papers being identified from the databases. 81 papers were initially judged to be relevant. Thus, the data analysis began with 81 papers, but some articles were excluded. Papers were excluded if the report was incomplete (e.g. the results were missing), or if it was exactly the same study as another one in the list (e.g. if an empirical study formed the basis for both a conference paper and an extension published in a journal). Finally, 53 papers were included in data analysis. In study 2, we found 109 papers initially.

After an analysis of the relevance, we were left with 74 papers. At the end, 42 papers were included in the data analysis. Papers were removed based on the same criteria in both studies, and hence the main difference is the initial way of finding papers, i.e. the search strategy.
45 paper were the same in the initial set of papers identified, 534 papers in study 1 and 109 papers in study 2. This overlap was surprisingly low. However, the situation changes when we look at the papers in the next step, i.e. those initially judged as relevant. In this step, 41 papers were identical, which should be compared with having 81 papers in study 1 and 74 papers in study 2. In the final set of papers, 27 of the papers were in common between study 1 and study 2. Figure 4.1 visualizes the overlapping papers at the last stage. All stages are summarized in Table 4.1, where the unique papers in each study are shown separately and the papers in common are shown as well.

### Table 4.1. Number of papers in two different studies

<table>
<thead>
<tr>
<th>Study No.</th>
<th>Initial Papers</th>
<th>Relevant</th>
<th>Analyzed</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>489+45</td>
<td>40+41</td>
<td>26+27</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>64+45</td>
<td>33+41</td>
<td>15+27</td>
<td>15</td>
</tr>
</tbody>
</table>

There is a huge difference between the numbers of papers we initially found (109 vs. 534), but it should be noted that we have checked the title of a lot of sources in snowballing too, i.e. when browsing the reference lists of the papers identified. The latter makes it hard to compare the numbers in the first step exactly. Based on the numbers in the table, it can be seen that there are 27 papers in common in the final set of papers, i.e. a slight majority of the identified papers is common between the two studies (study 1 and study 2). More details can be found in Appendix 4.1, Appendix 4.2, and Appendix 4.3 respectively. Papers in common are denoted with a B and they are listed in Appendix 4.1. Unique papers for study 1 are denoted with an F, and they are listed in Appendix 4.2. Finally, unique papers for study 2 are denoted with an S and listed in Appendix 4.3.

![Figure 4.1. Venn diagram for the overlapping papers](image)

A study such as [S6] with the title “cross-continent development using scrum and XP” could definitely not be found in the searches in databases, since in study 1 we did not add “cross-continent” in the search terms. The term was not used since we did not realize that we ought to identify it as a synonym to global software development. But it was found in study 2 (snowballing approach) because only by seeing these words in the title of the article, we could immediately recognize that “cross-continent” means global or distributed in this case. However, for this specific paper, the reference list did not add any further relevant papers. The example illustrates the challenges in formulating search strings while in snowballing it may become evident to the researcher to include a paper when reading the title of a paper.
4.4.2. Distribution of Papers

The next step in the comparison is to compare the distribution of papers across the years.

4.4.2.1. First Comparison

As mentioned above, the first comparison includes all papers found in both studies, while the second comparison (see below) compare the unique papers in each of the two studies. As shown in Table 4.2, the number of papers found in study 1 and study 2 in each year (from 1999 to 2009) is not the same.

<table>
<thead>
<tr>
<th>Year</th>
<th>Study 1</th>
<th>Study 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2001</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2002</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2003</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>2004</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>2005</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>2006</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>2007</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>2008</td>
<td>20</td>
<td>13</td>
</tr>
<tr>
<td>2009</td>
<td>17</td>
<td></td>
</tr>
</tbody>
</table>

However, the pattern of distribution does not seem to be completely different and both indicate that the number of papers has grown in the past decade.

4.4.2.2. Second Comparison

The number of unique papers found by each study in each year is presented in Table 4.3. We have found no unique papers in study 1 before 2004. Considering the number of papers in 2009, it is hard to conclude that the number of published papers is increasing in the past decade. In study 2, the number of published papers seems to be constant over the years.

<table>
<thead>
<tr>
<th>Year</th>
<th>Study 1</th>
<th>Study 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2001</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2002</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2003</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>2004</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2005</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2006</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>2007</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>2008</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

However, we should mention that this comparison is not fully fair because the total number of papers shall be compared against each other instead of considering only the unique ones. The question is whether the differences are due to the different search strategies.

4.4.3. Distribution of Research Types

Next, we wanted to compare the research types using the classification from Wieringa et al. [14]. In summary, the types are defined as follows:

- **Evaluation Research**: Techniques, methods, tools or other solutions are implemented and evaluated in practice, and the outcomes are investigated.
- **Validation Research**: A novel solution is developed and evaluated in a laboratory setting.
- **Solution Proposal**: A solution for a research problem is proposed, and the benefits are discussed, but not evaluated.
- **Conceptual Proposal or Philosophical Paper**: It structures an area in the form of a taxonomy or conceptual framework, hence provides a new way of looking at existing things.
• *Experience paper:* It includes the experience of the author on what and how something happened in practice.

• *Opinion Paper:* The personal opinion on a special matter is discussed in an opinion paper without relying on related work and research methodologies.

### 4.4.3.1. First Comparison

Both studies have found a majority of papers to be reported as experience reports in which practitioners have reported their own experiences on a specific issue and the method applied to alleviate it [14]. It should be noted that the number of papers in study 1 and study 2 for each research type is different. This difference is, however, expected due to the fact that the number of papers found in study 1 and study 2 are different.

In addition, the order of research types according to their frequency is different. The order in study 1 is: 1) experience report, 2) evaluation, 3) opinion, 4) solution, 5) validation and 6) philosophical, and in study 2 it is 1) experience, 2) validation, 3) evaluation, 4) solution, 5) philosophical and 6) opinion. It is surprising that we found no opinion paper in study 2 while it was the third most frequent research type in study 1.

### 4.4.3.2. Second Comparison

When it comes to a comparison of the unique papers, the majority of the current research was found to be in form of experience reports in both studies. And in addition to the identified research types in study 1, a solution paper is found in study 2.

### 4.4.4. Countries Involved in GSE

It is also possible to compare the most common combinations of collaboration. The combinations include both global collaboration and distributed development.

#### 4.4.4.1. First Comparison

In both studies, the collaboration between USA-India is found to be the most popular, and then distributed development within USA although the exact numbers are different.

#### 4.4.4.2. Second Comparison

The same pattern as in the first comparison is found through the second comparison.

### 4.4.5. Most Efficient Practices

To identify the most efficient Agile practices used in a GSE setting was one of the main objectives of the literature review, and hence this is an important aspect to compare. If the studies identify completely different Agile practices, then the search strategy has indeed influenced the outcome of the literature review.

#### 4.4.5.1. First Comparison

Considering the frequency of Agile practices in literature, we sorted the list of reported practices in both studies, where frequencies were counted based on the number of publications referring to a practice as being successful.

We classified the practices based on their rank in a descending list. For example, if the highest frequency was found to be 18 for practice A, then practice A was assigned to class 1 together with all other practices having a frequency of 18. It means that the rank of each practice, in the sorted list, was considered as the class the practice belongs to. The practices with the same frequency have been assigned to the same class. The purpose of the classification was to be able to make a fair comparison, since the number of analyzed papers was different in study 1 and study 2. The result of the comparison is summarized in Figure 4.2 (the x-axis represents the classes for the practices).
We would like to emphasize that it is not our intention to discuss the actual outcome in terms of which Agile practices are most efficient in a GSE setting. Our objective is to compare the outcomes from a search strategy point of view. We would like to refer anyone interested in the actual outcome regarding agility and GSE to Chapters 3 and 6.

If we take the first 3 classes of the practices, study 1 reported “standup meetings” (class 1), “sprint/iterations” (class 2), and “continuous integration” (class 3). Study 2 has found “sprint/iterations”, “standup meetings”, and then three practices are included in class 3 for study 2: “pair programming”, “sprint review/demo” and “test driven development” respectively. Thus, some practices are in common in the top and overall we can see from Figure 4.2 that the patterns are quite similar on the lower classes. The classes in the top part of Figure 4.2 are infrequent, i.e. belonging to classes 9 and higher. This means that these practices are mentioned by few studies and hence the difference is simply due to random, i.e. whether specific papers mentioning these practices are included or not. In general, the agreement of the most efficient practices is high, i.e. the lower classes in Figure 4.2.

### Figure 4.2. Agile practices in study 1 and study 2 – the first comparison

#### 4.4.5.2. Second Comparison

In this comparison, both studies found 18 practices, and 13 of them were the same. Five unique practices were found in each study (summarized in Figure 4.3). Once again, the highly ranked classes (low numbers) show a very strong agreement.
So far, we have presented and discussed which Agile practices were found most efficient in study 1 and study 2 as well as the countries involved in each combination of Agile method and distribution setting.

For the purpose of comparison, we have assigned different scores to the combinations. If the practices and the countries found are the same for the Agile GSE combination in study 1 and study 2, we have assigned the score 4 to the combination, score 3 if only the practices are the same, score 2 if only the countries are identical, score 1 if neither practices nor countries are the same and finally score 0 if the combination does not exist in the other study.

In this comparison, a majority of combinations are completely different. However, if we exclude the combinations, which were found in only one study (i.e. XP–open source in study 2 and Agile–open source in study 1), similar finding were identified for a majority of the combinations in both studies. Thus, it is clear that the comparison is very sensitive to individual studies.

Unlike the previous comparison, most combinations seem to be completely different. However, it may be due to that the comparison id very sensitive to individual studies, and given that we have fewer studies here than in the first comparison, it may make it even more sensitive. It should be noted that the combinations of Agile with offshore, open source, and virtual team were found only in study 1 whereas XP–Open source, and XP–Unclear were found only in study 2. The latter combination means that the setting was unclear, although XP was used.

As Figure 4.4 illustrates, we found exactly the same pattern (i.e. both Agile practices and the involved countries are the same in study 1 and study 2) for three distinctive combinations in the first comparison and only one combination in the second comparison. However, the number of combinations, which have the same score in both comparisons, is 6 out of 14, which implies 42% overlap. In addition, if we exclude the unique combinations, 6 out of 9 combinations have the same rank, which is 66% overlap.
In order to assure the reliability in this study, we tried to improve the reliability of the two systematic review studies in the first place (more details on this for each individual study can be found in Chapter 3 and Chapter 6). Then considering the purpose of this study, which was to conduct a comparison, we tried to perform the comparison as fairly as possible.

Therefore, the researchers were the same in study 1 and study 2, and the analyses on the data were performed as similar as possible. Although we had more experiences of doing systematic reviews when we started study 2, we tried to keep the gap between conducting searches as small as possible.

So the data was collected with a few months difference for study 1 and study 2, although we synthesized them later for study 2. The latter was done to ensure that the researchers did not remember all details when deciding on, for example, inclusion and exclusion of paper in the second study. In addition, two researchers were involved in reviewing the comparisons and drawing the conclusions of this study.

In addition, the order of conducting the studies might have affected the results of each study and as a consequence the result of the comparisons.

4.5. Discussions

4.5.1. Time and Effort Required

We cannot claim that SLRs are more time consuming in formulating the search strings because snowballing requires some formulations too. However, SLRs require separate formulation for each database whilst snowballing does not explicitly require searching in more than one database.

However, in the SLR study, we found much more noise compared to the snowballing study. Therefore, it consumed much more time and effort to refine the searches as well as identifying the relevant papers and discarding the irrelevant ones.
4.5.2. Noise vs. Included Papers

Considering the term “Agile” which is a very general word and used in many papers in many disciplines, we found a lot of noise in the SLR. Due to this, we had to limit the search to abstract, title, and keywords. But still the number of irrelevant papers was much higher than the number of included and analyzed papers. In the snowballing search, the balance seemed to be more reasonable.

4.5.3. Judgments of Papers

In snowballing, most of the judgments were done based on the title of the paper when going backward through the reference lists (or forward in the citations if being applied). In some cases, we judged papers once more based on their abstract, i.e. it resulted in performing a stepwise judgment. In the SLR, the judgments were done on title and abstract at the same time.

It should be noted that the papers with no relevant keywords in the title might be missed in snowballing. On the other hand, the papers that use different wordings (as in the example with cross-continent) might not be caught in an SLR.

4.5.4. Prior Experience

Prior experience of the researcher in the area of the studies as well as in performing the secondary studies may affect the results.

The differences can be seen, for example, when judging the relevancy of the papers. An experienced person already knows several papers and knows several active researchers in the area, which may affect the reliability of the secondary study regarding the relevancy of included papers.

4.5.5. Ease of Use

We found the snowballing approach to be more understandable and easy to follow in particular it is believed to be easier for novice researchers. The SLR provides a lot of guidelines, which is good on the one hand, but on the other hand novice researchers might find it confusing rather than helpful.

4.5.6. General Remark on Literature

It should be noted that a general problem with systematic literature reviews in software engineering is that in many cases existing papers are hard to classify and analyze. We have observed that many papers (in particular industrial experience reports) have insufficient information regarding the context and hence it becomes difficult to synthesize evidence from some studies.

4.6. Conclusions

In this paper, we evaluated two different first steps for conducting systematic literature studies. This was done by comparing two secondary studies on Agile practices in GSE, which were performed by the same researchers but using different search methods. First, we compared the studies against each other whether the same set of papers was found and if the included papers had resulted in the same conclusions. Secondly, we excluded the common papers from both studies and performed new analyses with the remaining unique papers for each study.

Considering the fact that these comparisons did not indicate any remarkable differences between the two different studies, we compared the actual results found using the two different search methods applied. A summary of the findings is provided below.

After comparing the two secondary studies in two different ways (with the common papers and with only the unique papers in each study), we did not find any major differences between the findings of the analyses. The figures and numbers were not the same, but the general interpretation of them is quite similar. We can summarize our findings as follows for the two research questions.
**RQ1: To what extent do we find the same research papers using two different review approaches?**

To answer the first research question, we may observe that the papers found are different both in the number and the actual papers. In addition, the final set of papers, which was used in data analyses, was also found to be different, although 27 papers were common. This is not really surprising given that we only used the first step in the two search methods, i.e. according to the different guidelines used. It is highly likely that the overlap would increase if we conducted snowballing from the papers found in the database searches, and also if we did forward snowballing when starting with backward snowballing. However, it is should be noted that a majority of the papers are the same despite only comparing the staring point for the comparison, i.e. database search vs. backward snowballing.

**RQ2: To what extent do we come to the same conclusions using two different review approaches?**

The answer to the second research question is more important, since it concerns the actual findings. Regardless of the differences in the actual numbers and figures, similar pattern were identified in both studies and hence similar conclusions were drawn. However, when excluding the same papers from both studies and analysing only the remaining unique papers of each study, the identified patterns seem to be slightly different, which may be due to having fewer papers (a smaller sample). Therefore, it is not easy to draw any general conclusions with respect to the second research question.

However, given the overlap, despite only conducting the first part in the guidelines, it indicates that the actual conclusions are at least not highly dependent on whether using database searches or snowballing. It is also quite obvious that the overlap will become larger if combining the two search strategies, although the downside being that it generates more work. Systematic literature studies are quite time consuming.

Snowballing might be more efficient when the keywords for searching include general terms (e.g. Agile), because it dramatically reduces the amount of noise. Our personal experience confirms this. However, we recommend applying both backward and forward snowballing.

Although these conclusions, recommendations, and findings are based on our experiences with this comparison study as well as previous secondary studies, they seem to be in alignment with some previous studies [6] and [8], but contradictory with some others [7]. In anyway, more such comparison studies are required to be able to compare the methods fairly.

**Acknowledgements**

This research was funded by the Industrial Excellence Center EASE – Embedded Applications Software Engineering, (http://ease.cs.lth.se).

**References**


Appendix 4.1. Common papers in study 1 and study 2


Appendix 4.2. Unique papers in study 1


Appendix 4.3. Unique papers in study 2


Chapter 5

Investigating the Applicability of Agility Assessment Tools: A Case Study

Abstract

Agile software development has become popular in the past decade whilst it is not specifically defined how much Agility would be sufficient in a particular situation. Although few conceptual frameworks exist in the research literature to evaluate the level of practicing Agile, no strong evidence exists that practitioners have utilized them. Hence, we searched the web for existing commercial tools and found several of them in forms of questionnaires/surveys, which are rarely reported in the literature. After exploring them, three tools were selected as input to a case study to compare their assessments of two Agile teams with their own and customers’ perceptions of their Agility. The paper describes the steps undertaken in this research and discusses the applicability of the studied tools by investigating their strengths and weaknesses.

Keywords

Agile, Assessment, Measurement, Tool, Survey.

5.1. Introduction

Agile values and principals in striving for more efficient approaches to develop software were initially formulated from a practitioner’s point of view [25]. Agile software development has received more attention in recent years and different Agile methods have been utilized in software organizations. Each method consists of several practices, however, the extent to which practitioners utilize the practices is not yet fully investigated.

This has motivated researchers to study what practitioners exactly mean when claiming being Agile through examining their level of adhering to Agile values, principals, and technical practices. Therefore, several frameworks with the purpose of assessing or profiling Agility have been developed (e.g. [14][5][6][34]). Although some of these frameworks are developed through empirical studies, to the best of our knowledge only the participating organizations have used them afterwards. The major reason could be that practitioners view them as context specific so that they cannot apply them without the help of Agile experts.

On the other hand, several questionnaires (e.g. [4][9][10][11][12][13][30]) have been developed and used by practitioners that have not been investigated by researchers. This indicates that researchers and practitioners seem to evolve Agile software development in parallel to some extent.

Therefore, we have searched the web for existing tools that are not found in the research literature. A tool is a set of questions (e.g. a survey or questionnaire) and an analysis on the responses to the questions. Using a selection of tools, we conducted a case study to evaluate their applicability through comparing their results with the perceptions/expectations of the participating teams and their customers.
After studying a number of commercial tools and discussing them with practitioners, we chose four tools (i.e. [11][12][13][16]) as particularly interesting to be further evaluated. They were chosen based on their coverage of different perspectives of Agility (e.g. team and organization) as well as different areas (e.g. requirements, testing and communication). The survey questions formed the basis for the data collection of this research and the results of the surveys provided the basis for discussions with the participating teams.

The remainder of the paper is structured as follows. Section 5.2 gives a brief background and summarizes related work. Section 5.3 discusses the research methodology and explains different steps of conducting the research. The results are presented in Section 5.4, and discussed in Section 5.5. Finally, conclusions and future research directions are presented in Section 5.6.

5.2. Background and Related Work

The concept of Agility as “flexibility” and “leaness” [32] in software engineering was introduced by practitioners [25] to mitigate the limitations of traditional software development approaches such as heavy documentation, long time to market, etc. Agility is also defined to be continuously receiving feedback and making changes in the software rather than rejecting the higher rates of changes [2].

Different variations of Agile approach exist, the most common ones are Extreme Programming (XP), Scrum, Feature Driven Development, Dynamic Systems Development Method, and Lean development [23][21]. Most Agile methods, however, encourage frequent and continuous face-to-face communication, customer feedback and requirements gathering, as well as pair programming (PP), refactoring, continuous integration (CI) and minimal documentation [23].

Despite the popularity of Agile methods, there is a lack of understanding to what extent different practices need to be used to become more efficient in software development, and which practices are actually applied in Agile or Lean software organizations or teams.

The related work in this area both in the research literature and in practice is briefly presented in the following.

5.2.1. Status of Research

Several frameworks, methods, or guidelines have been proposed for assessing, measuring, or evaluating Agility with the purpose of assisting software organization in the process of Agile adoption. A brief summary of the related research is given below.

In 2003, Abrahamsson et al. proposed an analytical framework to analyze the existing Agile methods [24] by introducing “analytical lenses” such as lifecycle coverage, project management, abstract principles vs. concrete guidance, universally predefined vs. situation appropriate, and empirical evidence.

In 2004, Barry Boehm proposed a five-step risk-based software development method by combining the characteristics of both Agile and plan-driven approaches [7]. It, however, does not guide practitioners what specific practices to follow.

Later, Williams et al. provided a benchmark measurement framework for assessing XP practices adopted in software organizations [33] which is composed of three parts: 1) XP Context Factors to record essential contextual information, 2) XP Adherence Metrics to concretely and comparatively express the utilized practices, 3) XP Outcome Measures to assess the team’s outcome when using a full or partial set of XP practices.

Conboy and Fitzgerald proposed a framework of software development Agility thorough review of Agility across many disciplines [32], and evaluated it in software development context through reviewing the relevant research over the past 30 years. They conclude that the Agile Manifesto principles do not provide practical understanding of the concept of Agility outside the field of software development.

In 2006, Hartmann and Dymond collected some of the current thinking on appropriate Agile metrics, and proposed simple tools to be used by teams or organizations [3] with the purpose of encouraging metrics that are more in alignment with the objectives of Agile teamwork.
Then, Qumer and Henderson-Sellers proposed 4-DAT, which is a framework-based assessment tool for the analysis and comparison of Agile methods [17]. It provides evaluation criteria for the detailed assessment of Agile software development methods through defining four dimensions: 1) method scope characterization, 2) Agility characterization, 3) Agile values characterization, and 4) software process characterization.

In 2007, similar to CMM and CMMI, Sidky Agile Measurement Index determines five Agile levels by measuring the number of Agile practices adopted by an organization [5]. The objectives for each level are set relatively to Agile values and principals as stated in the manifesto, and a set of practices is suggested for each level. However, enforcing these predefined practices is not in alignment with the flexibility of Agile methods.

A similar framework to the Sidky’s exist that measures flexibility, speed, leaniness, responsiveness, and learning in order to examine the Agility [1]. Here, the number of levels is six, and in addition, they are grouped into three Agile blocks. The level of Agility is determined for each block by analyzing the adoption of a set of predefined practices. It should be noted that both frameworks in [5] and [1] try to quantify the degree of practicing Agile rather than evaluating the effectiveness of the applied Agile method.

Hossain et al. in 2009 proposed a conceptual framework based on the research literature to address the challenges of combining Global Software Development (GSD) and Agile methods [34]. It is expected to assist project managers in deciding which Agile strategies are effective for a particular GSD setting considering the contextual information.

Furthermore, Taromirad and Ramsin introduced the Comprehensive Evaluation Framework for Agile Methodologies (CEFAM) in [14], aiming at covering all different aspects of Agile methodologies.

Soundararajan and Arthur [6] have proposed the Objectives, Principles and Practices (OPP) framework, in which the “goodness” of an Agile method is assessed by evaluating its adequacy and effectiveness, and the capability of the organization to provide the supporting environment. It means that for a given set of objectives for an Agile method, it has to be ensured that the appropriate principles and the proper practices are set.

Yauch constructed an objective metric for Agility performance that measures Agility as a performance outcome and captures both organizational success and environmental turbulence [26]. The metric was developed as a theoretical model and then validated through literature review, case studies, and pilot survey data.

5.2.2. Status of Practice

In order to assess the Agility of the process, different tools exist that basically examine the presence or absence of Agile practices. Some commonly used checklists by Agile practitioners are (1) the Nokia Test [30] for Scrum method, (2) How Agile Are You (42-Point Test) [9], (3) the Scrum Master Checklist [4], and (4) the Do It Yourself (DIY) Project Process Evaluation Kit [18]. It should be noted that most of these checklists are tailored to one or a few specific Agile methods.

Another concern for software organizations is to figure out how efficient the Agile adoption has been, which implies identifying problem areas and taking proper actions to address them. For this purpose, retrospective meetings at the end of each iteration are held to “fine-tune” the Agile practices. Furthermore, external consultants or tools can also help the assessment. Most assessment tools are, however, not publicly available free of charge, and therefore, information about them is also hard to find. In the following, we briefly discuss four different Agile assessment tools, which can be accessed for free on the web.

5.2.2.1. Comparative Agility

The Comparative Agility (CA) assessment tool is developed based on the assumption that most organizations prefer to be more Agile than their competitors rather than striving to be “perfectly Agile” [20]. Therefore, CA assesses the Agility of teams/organizations relative to other teams/organizations that responded to the questionnaire.
Chapter 5 – Investigating the Applicability of Agility Assessment Tools: A Case Study

CA is a survey-based tool, in which answers are recorded on a five-point scale. Questions can be answered from four different views, which are team, department, division, and organization. Eight dimensions are considered as the basis for the assessment: 1) teamwork, 2) requirements, 3) planning, 4) technical practices, 5) quality, 6) culture, 7) knowledge creating, and 8) outcomes [16].

Each dimension consists of three to six characteristics, and each characteristic of a number of statements that have to be verified by the respondents. Each statement indicates an Agile practice, and the respondent should determine the truth of the statement based on their actual practices in the team/organization.

Participants get a free report indicating their level of practicing Agility in each dimension and each characteristic separately in comparison to other practitioners.

5.2.2.2. Thoughtworks Agile Self Evaluation

The Thoughtworks Agile assessment survey is developed by Thoughtworks, which is a leading Agile development/consulting organization [11]. The survey consists of 20 multiple-choice questions addressing Agile development and management practices. Practitioners can complete the surveys online and get a report on the level of Agility within their organization/team, as well as improvement opportunities [11].

5.2.2.3. Thoughtworks Build and Release Management Assessment

The second survey developed by Thoughtworks is developed to assess build and release management practices. It consists of 20 multiple-choice questions [12]. In comparison with the other survey of Thoughtworks, it is focused on the build and release management activity rather than general Agile practices.

5.2.2.4. Business Agility Assessment

This tool provides a high level assessment of an organization by asking 15 questions [13]. The answers result in the organization being on one of the following five levels: initial, repeatable, defined, managed, or optimizing. The results display both business Agility score and the cumulative average of every completed assessment in the database. Available online: http://businessagility.btmcorporation.com. This survey is an appropriate candidate for assessing Agility on an organizational or business level.

5.2.3. Motivation

Most of the existing frameworks in the research literature do not address all Agility issues in their criteria, and only partially cover the usage context. However, the most important critique is their practical applicability since they cannot be used without help of a consultant.

On the other hand, Agile researchers have not evaluated the available checklists and tools developed by practitioners. However, it should be noted that these tools primarily evaluate the presence or absence of practices rather than the extent of being practiced.

Unlike other studies in the area, in this research, we looked for available commercial tools firstly and evaluated them based on covered Agile areas and their comprehensiveness. The selected tools were utilized as input to an industrial case study with the purpose of evaluating their applicability.

In this research, instead of introducing a new framework, we scientifically examined the commercial Agile assessment tools. This helps both researchers and practitioners to gain awareness of the existing work in the area and to benefit from an analysis on the strengths and weaknesses of the studied tools.
Chapter 5 – Investigating the Applicability of Agility Assessment Tools: A Case Study

5.3. **Research Methodology**

The research started by searching the relevant literature and the web for current frameworks, or tools with the purpose of evaluating the Agility of a software development team or organization. Since, we did not find the existing frameworks sufficiently testable (e.g. through a questionnaire that could be used in a case study), we focused the searches solely on the tools that are developed and used by practitioners.

Therefore, the major purpose of the research was to investigate the applicability of the existing commercial tools for assessing the Agility of a team or an organization. The research questions (RQ) were formulated as follows.

RQ1: Which commercial tools exist to evaluate the Agility of a team or an organization?

RQ2: Are the existing tools applicable to assess Agility?

RQ3: Do the existing tools give the same assessment results?

In order to answer RQ1, we searched the web and preliminary evaluated the tools and selected a few of them as input to a case study to investigate RQ2. Finally, RQ3 was answered by comparing the results of the tools examined in the case study.

The research methodology was a mixed approach [27] applying both qualitative and quantitative data collection and analysis methods in an industrial case study. The undertaken steps can be summarized as follows.

5.3.1. **Preparation**

The preparation phase consisted of finding existing commercial tools and preliminary evaluating them in order to minimize the time required from industrial participants. Here, we only summarize the tools actually chosen to use in the case study.

5.3.1.1. **Searching the Web**

In the Google\(^1\) search engine, we initiated the searches by formulating the search string as: Agile AND (assess OR measure OR framework OR tool OR evaluate OR profile). That resulted in enormous number of hits (i.e. about 323000). Therefore, we tried simpler search strings such as “Agile assessment tool” or “Agile measurement survey”. However, the search string that seemed to give more reasonable results was “Agility assessment tool”. In the following section, we briefly describe each tool and discuss its strengths and weaknesses as it was performed in the initial evaluation step of this research.

5.3.1.2. **Initial Evaluation of the Tools**

The first criterion was to be able to access the tool for free. However, we also examined the ones that provide trial version. In addition, they were further examined based on their questions and the covered areas regarding the Agility.

Another important factor was the presentation of the results. Therefore, we gave random answers to all questions of available tools and explored the results in order to figure out how they are being extracted.

5.3.1.3. **Nokia Test**

It examines to what extent the team follows Scrum practices [30]. It consists of nine questions regarding: iterations, testing, Agile specification, product owner, product backlog, estimates, sprint burn down chart, team disruption, and team. Available online: [http://antoine.vernois.net/scrumbut/?page=test&lang=en](http://antoine.vernois.net/scrumbut/?page=test&lang=en).

The test is simple and time efficient, but it covers only Scrum practices rather than whole picture of Agile development. Furthermore, it helps when forming a Scrum team rather than being a tool for assessing the status of the team in different points of time.

\(^{1}\) [http://www.google.com](http://www.google.com)
5.3.1.4. 42-Point Test

This test provides a set of statements to assess the team’s adherence with Agile principles and methods [9]. Each team member answers the questions separately and if the team is consistent they should get the full score (i.e. 42). Available online: http://www.allaboutagile.com/how-agile-are-you-take-this-42-point-test.

This test is developed inspired from the Nokia test [30], but considers Agile principals instead of only Scrum practices. However, similarly it can be used as a checklist to assess presence or absence of a practice rather than evaluating the level of applying them.

5.3.1.5. Thoughtworks Studio’s Assessments

Thoughtworks Studio has developed two different surveys. One is more abstract and focuses on Agile practices, and the other one is on a more detailed level and contains questions related to build and release management activities.

**Agile Self Evaluation:** It consists of 20 questions covering management and development practices [11]. Management practices are related to requirements analysis, business responsiveness, collaboration and communication, project management, and governance. Development practices address simplicity, build management, configuration management, and testing and quality assurance. Available online: http://www.agileassessments.com/online-assessments/agile-self-evaluation.

Although the questions are related to different Agile areas and perspectives, the number of questions per area is not sufficient for all categories. However, the definition of Agility and the highest level of practice are clearly defined and are transparent to the users. In addition, the results are presented visually along with relevant improvement recommendations for each Agile area.

**Build and Release Management Assessment:** This test has 20 questions addressing build management and continuous integration, testing, data management, deployment and environment management, release management, and configuration management [12]. It helps team to get software releases right in the first set up. Available online: http://www.agileassessments.com/online-assessments/brm-self-evaluation.

Similar to the before mentioned survey, the major problem is too few questions per area, in which the potential risk is that one question dominates the result of an area.

5.3.1.6. Scrum Checklist

This checklist is helpful when setting up a Scrum team in order to ensure existence of required practices [28]. Available online: http://www.crisp.se/scrum/checklist. Besides its simplicity, it does not fulfil the evaluation criteria of this research.

5.3.1.7. Business Agility Assessment

This tool helps assessing Agility on an organizational or business level [13], and it was elaborated on Section 5.2.2.4.

5.3.1.8. CMI Lean Agility

This approach calculates the company’s health or lean Agility for different business units, sites and/or business processes [15]. Available online: http://www.cimes.be. This tool is not available for free on the web and hence was removed from our list of investigation.

5.3.1.9. Signet Consulting

This test provides 22 questions as a quick self-assessment of the Agility of an organization, a team, a department, or a division [29]. The answers vary from 1 to 5 based on the level of applying the practices. It should be noted however that the focus of this test is on the organizational Agility. Available online: http://www.signetconsulting.com.

This small test is in particular helpful when evaluating the organizational Agility. However, its presented analysis is not sufficient for the purpose of this research.
5.3.1.10. Comparative Agility Assessment

CA is a survey-based tool that assesses the Agility of teams or organizations relative to other teams/organizations that responded to the questionnaire up to date [16]. The questions sufficiently address different areas of Agility and the number of questions per area is reasonable. It is possible to answer the questions from different views that are team, department, division, and organization. Furthermore, the provided analysis is visual and comprehensive.

5.3.1.11. Dr Agile

It measures organizational readiness to adopt Agile practices that is suggesting suitable practices for the organizational setting and/or highlighting required improvement in the practices [19]. This test consists of 22 questions, which shall be answered on behalf of the team. However, it is not available for free on the web and hence was removed from our list of further investigation.

5.3.1.12. Agile Karlskrona Test

It is a generic Agile adoption test, which is developed based on Agile principles [10]. It assesses a software development team through eleven simple multiple-choice questions, and presents the team’s score regarding practicing Agility. Available online: http://www.piratson.se/archive/Agile_Karlskrona_Test.html. This test can be used for monitoring a team’s transition from non-Agile to Agile, but it is not sufficient to assess/profile Agility of an Agile team.

5.3.1.13. Summary

As discussed in previous sections, among all studied tools, we selected four of them for further investigation in a case study: 1) Survey 1 (S1): comparative Agility assessment [16], 2) Survey 2 (S2): thoughtworks’ Agile self evaluation [11], 3) Survey 3 (S3): thoughtworks’ build and release management assessment [12], 4) Survey 4 (S4): business Agility assessment [13].

5.3.2. Design and Conduct

In this section, we explain the details of research design and conduct in relation to the case study, including the questionnaire, interviews, and open discussions.

5.3.2.1. Questionnaire

All questions from the four selected surveys were merged in one Excel file (in separate sheets). In addition, some questions were added to capture the demographic information of the participants (e.g. role and years of experience) as well as project information (e.g. type and customer).

For each question in all surveys, we added the option to skip answering if it was not applicable or the respondent did not want to reply to it for any reasons. Furthermore, participants were supposed to declare how sure they were of the given answer for each question. The reason for this was to weight the answers with more confidence level higher than the answers with low confidence.

In addition, classification of the questions as it was in the original surveys was removed in order to reduce the bias when answering the questions. A sample of the survey is presented in Appendix 5.1.

At the time we designed the questionnaire, S1 consisted of 127 questions excluding the demographic questions, S2 of 20, S3 of 20, and Survey 4 of 15 questions. In total, the study includes 182 questions. Therefore, the time to finish the questionnaire was estimated to be maximum 180 minutes. All participants were informed about the purpose of the study as well as its potential benefits in advance.

During the conduction of the survey with the participants, we decided to omit Survey 4 mainly due to it not being perceived as applicable for a consultancy company. These companies provide services rather than products and therefore they perform market analysis and planning differently.
5.3.2.2. Interviews

Three different interviews were designed with the Scrum master, a team representative, and the customer, and are elaborated as follows.

5.3.2.2.1. Scrum Master and Team Representative

The major reasons for conducting interviews with the Scrum master and the team representative were to discuss the inconsistencies observed in the collected answers, and to gather additional contextual information of the organization, team, project, and the customer.

For this purpose, shortly after data collection, both researchers responsible for this study scanned the answers separately and found inconsistencies and prioritized them before bringing to the interviews.

The candidate questions to bring up at the interviews were identified as follows: 1) with no majority (if at least half of the participants did not give the same response), 2) with a big difference in the answers (e.g. “strong yes” vs. “strong no” for the same question), 3) questions that could dramatically change the score in specific areas, 4) with a big agreement, 5) with the majority of “Not Applicable” response.

They were prioritized in the same order too (i.e. the ones “with no majority response” were more important to be discussed and so on). The main motivation for prioritization was the time limit for the interview, which was agreed to be maximum 90 minutes. Since we had to collect additional contextual information, it was impossible to discuss important identified inconsistencies during the interview if they were not prioritized.

The team representative was nominated by the Scrum master and the company director. The same items were discussed with the Scrum master and the team representative intending to openly discuss the reasons causing the inconsistent responses. If both interviewees had given the same response (in the meetings), that was considered as the team’s response. Otherwise, we considered the question as “Not Applicable”. A sample of interview questions is presented in Appendix 5.2.

5.3.2.2.2. Customer (Product Owner)

In order to collect the customers’ perception on the team Agility, we designed a 60-minute interview with their customer representatives (separate customer interviews for Team 1 and Team 2). The purpose of the interviews was explained beforehand to the interviewees in an email.

Since the customers were not expected to be aware of all details about the teams in particular the technical aspects, we designed the interviews in a way to generally discuss the teamwork, team culture, knowledge creating, requirements, planning, technical practices, and quality. In addition, each interviewee was asked to assign a score from 0 to 5 (not Agile to highly Agile) to each before mentioned area.

5.3.2.3. Open Discussions

The results were presented to the whole teams in a retrospective meeting, and team members openly discussed if it matched their own perceptions. The teams actively took part in discussing the reasons for not being Agile in some areas, and how it can be improved or why it is an external factor that cannot be changed.

5.3.3. Participants

In the following section, we describe the participating organization and the teams taking part in the survey as well as their customers. The contextual information is reported according to the guidelines recommended by Petersen and Wohlin [8].
5.3.3.1. Softhouse Consulting

Softhouse Consulting was founded in 1996 as an independent IT consultancy company in Sweden, and currently is one of the leading Scandinavian suppliers of Lean Software Development with over 100 employees\(^2\). It has sites located in Malmö, Karlskrona, Stockholm and Gothenburg.

The company encourages simplicity, reliability, and professionalism in its work process and employs cross-functional and self-organizing teams to deliver fully functional software with the desired quality.

This study was conducted in cooperation with the development site located in Karlskrona, Blekinge, Sweden. Two teams participated in the survey, and their detailed information is given in the following.

5.3.3.1.1. Development Team (Team 1)

The team is responsible to add new features to an already existing system.

**Product:** The work is contract-based and the business relationship is onshore outsourcing [31]. The final product is commercial and is customized for specific telecom operator(s). At the time of data collection, the collaboration was built over 2.5 years through different projects, and each project is about 500 person-hours. The programming language is C++ and Java.

**Processes:** If the end customers are not satisfied with the standard features, they can order new features. The product owner prioritizes the orders applying first-in-first-out strategy. A project to add a new feature is normally about 200-400 hours.

Requirements (i.e. features) are small and therefore a number of small projects are run simultaneously within the team. One project is normally accomplished within four weeks (one sprint). The team receives a fixed set of requirements and a fixed deadline, and then independently performs Sprint planning and the design. However, it must handshake the design with the customer’s architect team to ensure its consistency with the whole product.

Although the team applies Scrum practices, the work process is adapted to the customer’s processes. For example, the practices within the team are Scrum, but the reports are made according to the customer’s preferences. The team follows coding conventions (which are accessible in the wiki). In addition, it applies automated testing and Test Driven Development (TDD). Daily meetings are held within the team to report the progress and discuss the issues, and weekly meetings with all personnel.

**Practices, Tools and Techniques:** Since the customer does not provide any Agile specific tools, Excel sheets are used for Sprint planning (e.g. to visualize the task loads, reprioritize the tasks, and provide burn down graphs to be sent to the customer as weekly reports).

**People:** One functional team shall consist of four to five people according to the customer’s demands. However, eight people are in Team 1 (T1) so that they can form two teams at the same time working on different tasks if needed. One person out of eight takes the role of system analyst, one is Scrum master, and six are developers/testers. The average prior experience of Agile (before this project) is about one year excluding the Scrum master and 1.2 years including the Scrum master.

**Organization:** The team is co-located at the customer site in the same city, but works independently. The team uses informal and face-to-face communication with no bookings.

**Market:** The project is developed for only one customer (although the customer would sell it to more than one other customer), and the studied organization’s strategy is to reduce the delivery time with good quality.

**Customer:** The customer organization is globally distributed and its software development teams have adopted customized Agile.

5.3.3.1.2. Maintenance Team (Team 2)

This team was formed 0.5 year after T1 to work with reported defects and maintenance issues (the same product as for T1). Most of contextual information is the same as for T1 and therefore we describe only the differences.

Processes: Any type of problems with the product found by the end users, market, or test departments, are documented as “trouble report” (TR). Team 2 (T2) receives TRs and runs trouble shooting i.e. repeating the error, finding where to fix it in the code, writing the test case, preparing the follow up forms, and sending the follow up to the test team.

The team has the freedom to plan only the tasks of a day, because the customer prioritizes the tasks, which TRs should be done first. So, the operation is somehow sprint-less. Regular meetings (three times per week) are held with the customer to collect the TRs and discuss the progress.

Practices, Tools and Techniques: A customized Kanban board is used which shows only backlog, team, and done columns to represent the plan and the workflow. Since the work is not really planned in Sprints, retrospective meetings have not been held regularly.

People: T2 consists of 10 people in total, two system analysts, one Scrum master, and seven developers/testers. The average prior experience of Agile is about 0.5 year excluding the Scrum master and 0.75 years including the Scrum master.

Customer: Although the customer organization is the same as for T1, the unit is different.

5.3.4. Data Analysis

Data analysis consisted of: calculating the mode for each question of the survey to represent each team’s answer, calculating the level of Agility based on the customer interview, and discussing each team’s Agility according to the studied tools.

Participants could determine how confident they are about the given answer for each question by selecting an option among: “sure”, “more sure than unsure”, “neither sure not unsure”, “more unsure than sure”, and “unsure” (i.e. to weight the answer between 1 to 0). In addition, it was possible to write comments.

Immediately after data was collected, we checked whether participants had provided all required information. Otherwise, we contacted them for clarification or to complete the questions not answered.

We summed up the confidence levels for each specific option/answer given for a question. The answer with the highest sum of confidence (mode value) was then considered as each team’s answer. It is explained in the following example. Suppose six team members have answered question 1 as it is shown in Table 5.1.

Table 5.1. Sample of given answers

<table>
<thead>
<tr>
<th>Person 1</th>
<th>Person 2</th>
<th>Person 3</th>
<th>Person 4</th>
<th>Person 5</th>
<th>Person 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sure</td>
<td>Answer</td>
<td>Sure</td>
<td>Answer</td>
<td>Sure</td>
<td>Answer</td>
</tr>
<tr>
<td>0.75</td>
<td>1</td>
<td>1.0</td>
<td>2</td>
<td>0.5</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.25</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Option 1 has 0.75+0.5+0 confidence level, option 2 with 1.0, and option 3 with 0.5+0.25 as it is shown in Figure 5.1. After weighting the answer with the level of confidence, answer number 1 becomes the team’s answer.
For some questions, the mode value was not identified because either more than one answer/option had the highest weight or less than the majority of the team members (i.e. \((\text{team size}/2)+1\)) had given an identical answer regardless of its weight. Those questions were brought to the discussions with the Scrum master and each team’s representative separately. If both gave similar explanation and selected the same answer, we considered that as the team’s answer, otherwise it was set to “not applicable”. Then, all answers were entered to the studied online survey tools and the results were presented and discussed with the team. This is elaborated in Section 5.4.

In the meeting with the customer, we discussed how they perceived each team’s Agility in each Agile area rather than asking detailed questions. The interviewee was asked to give a score from 0 to 5 (not Agile to highly Agile) to the following areas:

**Teamwork:** How much the team’s composition, management, communication is perceived Agile.

**Requirements:** This area includes the level of details of requirements, accommodating changes, technical design of requirements, and collaborating with the product owner.

**Planning:** It concerns the suitability of planning activities e.g. planning time, levels of planning and progress tracking.

**Technical Practices:** Technical practices are test driven development, pair programming, refactoring, continuous integration, coding standards, and collective code ownership.

**Quality:** This relates to automated unit testing, customer acceptance tests, and timing.

**Culture:** To what extent the customer views the team’s management style, response to stress, and the customer involvement as Agile.

**Knowledge Creating:** If the team learning is evident to the customer and is useful.

Each area was weighed the same (i.e. all are equally important) and the mean value was calculated to represent each team’s Agility level from the customer’s perspective. If the customer did not have enough information of a specific area, it was removed from the calculations. The results are more elaborated in Section 5.4.

5.3.5. **Summary**

The major objective of the research was to investigate the applicability of the existing commercial tools for assessing/profiling Agility. Therefore, we studied several tools in order to select a few of them as input to a case study. The case was the studied organization and two units of analysis were the two studied teams [22].

The participants of the case study were: 1) two software development teams in a consultancy company in Sweden that answered the questions of selected surveys, 2) Scrum master of the teams to clarify the answers of questions that required discussions, 3) representatives of each team with the same purpose as meeting the Scrum masters, and 4) customers of both teams to collect their perception of each team’s Agility. The extracted results of tools on the level of practicing Agility by teams were compared against the teams’ own perceptions of Agility, and their customer’s perception. The results of investigations are presented in the following sections.
5.4. Results

The results of each tool for T1, T2, and both are presented separately followed by customer’s perception. It should also be noted that one person is common in both teams, i.e. both teams have the same Scrum master.

5.4.1. Survey 1 (S1)

It considers Agility as the ability to adapt to ever-changing environments, and the questions are based on the Agile principles including both management and development practices. Based on the answers, it ranks the team/project on a scale ranging from “regressive” to “Agile”. Being in the “regressive” state represents behaviours that hinder the ability to adapt and the “Agile” state represents behaviours that focus almost exclusively on adapting changes.

When presenting the results to the team, we made minor modifications, which were changing “ad-hoc” to “neutral”, and “regressive” to “non-Agile” in S1 and S2, which sounds more positive and encourages the team to participate in the discussions around the results instead of risking a more defensive mode. Figure 5.2 summarizes the results of the survey for both teams separately and together.

During the presentation session, the teams also categorized the improvement areas as external and internal. The external refers to the factors that are introduced outside the team (e.g. due to the limitations on the customer site). The internal areas were listed down in the action plan of the team.

5.4.2. Survey 2 (S2)

Survey 2 helps to determine the level of applying Agile practices related to building and releasing software because these practices can reduce the release time from months to hours if utilized properly.

Figure 5.2 indicates that T2 is more Agile than T1 in both S1 and S2, while both together as a team stands in the neutral area in both surveys. As it shows, only T2 in S1 is Agile.

![Figure 5.2. Results of survey 1 and survey 2](image.png)

Surprisingly, both teams together in S2 are as Agile as T2. Considering the weighted mode response for each question as the majority might lead to domination of the answers with higher level of confidence, and hence this type of outcome. T2 team members might have been more confident about their given responses.

5.4.3. Survey 3 (S3)

This survey produces two different graphs based on the answers. The first graph shows the level of Agility in different areas/dimensions [16] in comparison to the other answers in the database. The second graph represents Agility level for all characteristics of each area. The analysis for each question is given in terms of the number of standard deviations from the average value in the. So, having a positive score means the answer is “better than” the average answers in the CA database. More details regarding the analysis can be found in [16].
Chapter 5 – Investigating the Applicability of Agility Assessment Tools: A Case Study

Figure 5.3 summarizes the results (only the first graph) for T1, T2, and all together. Surprisingly, T2 is ranked less Agile than T1 in this survey, which is different from S1 and S2. We will discuss the possible reasons for this contradiction in Section 5.5.

![Figure 5.3. Results of survey 3](image)

The results of three different surveys are summarized in Figure 5.4. T1 is Agile in S3 and “neutral” in S1 and S2. On the other hand, T2 is Agile in S1, and “neutral” in S2 and S3.

![Figure 5.4. Summary of the results](image)

5.4.4. Collective Agile Areas – All Surveys

We conducted an interview with the Scrum master after presenting the results to the team and discussing how it matched their own perceptions of practicing Agile as well as the reasons for not being Agile in relation to specific aspects. The main intention of the interview was to hold an in-depth discussion around the applicability of the studied surveys as well as the potential combination of them. The applicability in this context means to what extent it represents the actual status of the studied teams.

One critical topic of discussion was around the contradictions in the results of different surveys. For example, T2 is more Agile than T1 in S1 and S2, but less in S3. Furthermore, T1 is Agile in most areas of S3 while T2 is not, but all together becomes sufficiently Agile (Figure 5.3).
To examine the combination of surveys, we put all Agile areas from all surveys in a scale from “not Agile” to “very Agile” and discussed whether it depicts the teams’ status better than each individual survey. Figure 5.5 maps the studied Agile areas to the Agile scales. If one specific area was examined in more than one survey, we considered the one with higher Agility rank. It is surprising that T1 and T2 perceive project management (PM) differently although the same person (i.e. Scrum master) is responsible. It however might be due to the differences in their tasks and the way the work is formulated e.g. T1 uses sprints while T2 tasks are oriented differently to handle TRs.

![Figure 5.5. Collective Agile areas from all surveys](image)

5.4.5. **Customer’s Perception**

The customer representatives were interviewed in order to complete the contextual information of the project as well as to gather their view on the teams’ Agility. Figure 5.6 summarizes both teams’ scores (1 to 5) in different areas. It should be noted that T2 was not given any score in the “knowledge creating” area due to unawareness of the details. Therefore, it was not considered in calculating the mean Agile value for T2.

The mean value for the T1 was calculated to be 4.4 and 3.5 for T2. This is in alignment with the Scrum master’s perception.

Accepting these two perceptions, T1 is more Agile in practice than T2, which matches the results of S3. More in-depth discussions are provided in the following section.

94
Chapter 5 – Investigating the Applicability of Agility Assessment Tools: A Case Study

5.5. Discussions and Observations

The strengths and weaknesses of each survey is discussed separately with regards to: covered Agile areas, the number of questions per area, formulation/readability of questions, total number of questions, transparency of the analyses, presentation of the results, reputation of the survey developer, updates/reformulation of the questions or analyses, being acknowledged by academia, and finally on their additional features.

5.5.1. Strengths and Weaknesses of Surveys

Survey 1: Although the examined Agile areas seem to be sufficient, the number of questions per area is rather low. The potential risk is that some areas are assessed only by one question (e.g. “governance”). On the other hand, the total number of questions is 20, which can be considered as an advantage since it does not take long time to answer all of them. The participants had some difficulties in interpreting a few questions, but in general we cannot claim that they are ambiguous. For each question, different number of options are given that can be seen as a disadvantage if one plans to analyze the data besides the provided analysis by the tool.

The analysis is transparent and the presentation of the results is visual along with valuable recommendations. The organization providing the tool is well known and the tool has been recently mentioned in the research literature (e.g. in [1] and [14]).

Survey 2: S1 and S2 are developed by the same organization, and the advantages and disadvantages of S2 are similar to S1. But it is focused on build and release management practices rather than Agile practices. It is recommended to use both S1 and S2 for Agile assessment.

Survey 3: A sufficient number of Agile areas are examined with a rather high number of questions. The total number of questions is high in comparison to the previous surveys, but answering them is quite simple since one should determine to what extent given statements are correct (scoring them from true to false). Like previous surveys, it is hard to judge its clarity (there is always some possibilities to misinterpret a specific question, but this is valid for most surveys). This survey is mentioned in literature (e.g. in [1] and [14]) and has been updated several times in the past one year (between March 2011 and March 2012) (e.g. reducing the number of questions, reformulating the questions, etc). We believe that the main disadvantage with this tool is that the analysis is not transparent although the results are sufficiently well presented.

In S3, it is possible to ask for a separate link so that all team members could participate and give their own answers independently whilst this feature is not provided in other studied tools.
5.5.2. Comparisons of the Results

The Agility of two different teams was assessed through three commercial tools as well as interviews with the Scrum master and the respective customer for each team. In the following, we compare the results of the tools with each other and then with the perception/expectation of the teams, the customers, and the Scrum master. It should be noted that the results were not discussed with the customers and interviews were held separately with different set of questions.

5.5.2.1. Team 1

T1 is “neutral” in S1, “neutral” in S2 and “Agile” in S3. One reason for this contradiction could be that S3 is comparative and T1 could be more Agile than the teams in the database whilst not highly Agile by itself. In addition, the number of questions per area in S1 and S2 is smaller than in S3, which means if the participants had difficulty interpreting a question in S1 or S2 and answered wrongly, the rank could become “non-Agile” only by that question in that specific area whilst more questions were asked in S3 to judge the Agility.

Another type of contradiction in the results is having different scores in the same areas that are examined in the surveys. These areas are:

Testing: T1 is “non-Agile” according to S1 in “testing and quality assurance” but “neutral” in “testing” according to S2. One possible explanation is although the areas are named identical, the questions are quite different in the details.

Requirements Analysis: S1 ranks the team as “non-Agile” whilst it is “neutral” in the “requirements” area according to S3. The difference could be due to the number of questions that is 3 in S1 and 14 in S3. The greater number of questions could lead to the higher score since misinterpreting one or two questions out of three could lead to being “non-Agile” in S1.

Build Management: The team’s score in “build management” is “neutral” in S1, but “non-Agile” in “build management and continuous integration” in S2. This can be due to additional questions related to continuous integration in S2.

Configuration Management: This area is scored “Agile” in S1 while it is “neutral” in S2. Here, the only possible explanation could be different formulation of questions.

Collaboration and Communication: S1 says that T1 is “Agile” in “collaboration and communication” and S3 scores the team “Agile” in “teamwork” too. By comparing the questions, we realize that “collaboration and communication” in S1 is partially matched with “teamwork” in S3. However, it is scored the same in both surveys.

The team and the Scrum master believe that not being Agile in requirements analysis is not unexpected since the team’s influence on this process is minimized because it is completely handled by the customer. In addition, they think that they can be interpreted as “non-Agile” in “technical practices” too since they do not follow them exactly as in literature. However, they perceive themselves as Agile regarding the quality, testing, and communication. The customer has scored the team highly Agile in all areas and is satisfied with its working style and the performance.

5.5.2.2. Team 2

T2 is “Agile” in S1, “neutral” in S2 and “non-Agile” in S3. It is surprising that T2 is “Agile” in S1 but “non-Agile” in S3 because we observed the opposite for the other team. This could be due to the large number of “not applicable” answers given by the team members in S3, which could simply be caused by the order of surveys in the questionnaire. S3 questions were the last in the list of questions and participants could become tired or less motivated to respond to them carefully in comparison to S1 and S2 (supposing that one responds to the questions from the beginning to the end). However, we contacted the team members to confirm their responses, but did not receive any new responses.

The observed contradictions in “testing”, “requirements analysis”, and “configuration management” are exactly the same as for T1. However, the team is “Agile” in “build management” in S1, but “neutral” in “build management and continuous integration” in S2, which is slightly different than for T1. In addition, S1 scores T2 “Agile” in “collaboration and communication” and S3 “neutral” in “teamwork”, which is surprising.
Chapter 5 – Investigating the Applicability of Agility Assessment Tools: A Case Study

Both teams and the Scrum master were not surprised by not being scored Agile in “requirements analysis”, “technical practices”, and “knowledge creating”. They are with them being Agile in “quality”, “testing”, and “communication”. Their customer perceives the team Agile in all areas like for T1.

The third type of contradiction that we observed was different scores by the teams in the areas that are expected to be exactly the same mainly because they are either performed by the same people or in the same way. These areas are “project management” and “build management”. In the results, T1 is “non-Agile” whilst T2 is “Agile” in “project management” which pinpoints a big difference between the teams’ perceptions. Furthermore, T2 is “Agile” in “build management” but T1 is “neutral”.

Although the difference between “Agile” and “neutral” is smaller than between “Agile” and “non-Agile”, it shows misalignment in the teams’ perceptions/expectations.

T1 and T2 work in the same organization, the same field, for the same customer organization (but different units) and are managed by the same Scrum master. Having these similar conditions, we expect the teams to be equally Agile and aligned in the Scrum way of working. The differences however could be due to utilizing slightly different practices by the teams. For example, retrospective meetings are skipped by T2 while T1 regularly holds them. In addition, if the communication among team members is not effective, misalignment could be seen as we observed in this research study.

On the other hand, both teams were very open to the discussions and actively participated in interpreting the results of the tools, which is according to the Agile principles.

5.5.3. Candidate Survey Tool

As discussed in the previous sections, the studied tools did not assign the same scores of Agility to the studied teams. Therefore, we discussed the results with the participants in order to decide which survey or a combination of surveys is the best tool for assessing a team/project’s Agility (at least in our context).

We compared the results with the perceptions of the teams, their customers, and Scrum master. Generally, the perception is that the S3 results are more aligned with the collected perceptions for T1. However, its results for T2 are slightly different from the expectations. Furthermore, we discussed the differences between the teams with the Scrum master and figured out that S3 shows the differences better than S1 and S2.

The main advantage of S3 is that it provides more questions for each area in comparison to S1 and S2.

5.5.4. Threats to Validity

The validity threats regarding reliability and generalizability of this research as well as what we did to overcome them are discussed as follows.

Internal Validity: In order to draw valid conclusions, we applied a triangulation technique, which is a method that compares three or more types of independent perspectives on a given aspect of the research process (methodology, data, etc) [35]. The triangulations used in this study were data, investigator, and methodology triangulations.

Data Triangulation: The data was collected from three sources (team, Scrum master, customer) in order to capture different perspectives/expectations on Agility.

Investigator Triangulation: In data collection and data analyses, more than one researcher was involved in performing and validating the work.

Methodology Triangulation: We collected data both qualitatively (interviews and open discussions) and quantitatively (survey). Hence, both quantitative and qualitative data analysis were performed.

Furthermore, we studied two units of analysis (T1 and T2) in the case study rather than relying on single data points for a single team.
It should be noted that we did not transcribe the interviews immediately and did not directly confirm the content with the interviewees. However, the results of analyses and the conclusions were openly presented and discussed with all participants, which reduces the risk of misinterpretations and misunderstandings.

A concern with the results is that the analyses performed by the tools are not completely transparent. For example, the number of participants is small in comparison to the number of questions, and we are not aware how it is managed in S3 to ensure the reliability of the statistical calculations. In addition, the S3 questions are updated over time, which makes it hard for us to know how the new questions are compared against the existing database with different set of questions.

S3 provides a separate link that all team members can answer the questions independently, and it provides the collective results for the team rather than for each individual participant. We did not use this feature mainly because we intended to independently calculate the team’s representative answer. Furthermore, before inserting the responses to the tools, we discussed ambiguities around the responses in separate meetings with the purpose of clarifying the answers. The team’s representative answer for each question was finalized after discussions with the Scrum master and the team representative member.

We used the survey questions and options with no modifications and therefore any ambiguity with them remained. However, we added the option to skip a question as well as to provide the confidence level for the given answer in a way that the participants could determine how sure they are about the given answer rather than skipping it completely. Furthermore, the questions are grouped in the original surveys whilst we hid the classifications to the participants mainly to avoid bias when answering the questions. On the other hand, this could cause confusions and misinterpreting the focus of the question.

Optimally, from a research methodology point of view the order of questions in each survey and order of the surveys should have been random. We did not manipulate the list of questions in the surveys, but placed each studied survey in a separate Excel sheet so one could answer them in any desirable order, although most likely the respondents answered them in the same order.

Another threat is that one participant (the Scrum master) is in both teams. We performed separate analyses for the teams excluding the Scrum master, and concluded that the scores for T1 and T2 become less alike in this case. In this paper, we have not reported the results of separate analyses excluding the Scrum master.

Finally, due to other obligations of the participated teams, some of the team members could not take part in the case study. Nevertheless, 6 members out of 8 from T1, and 6 out of 10 from T2 indicate 75% and 60% participation respectively, which is more than average in both cases. However, the responses of other 4 members of T2 might have influenced the team’s score in different Agile areas and hence the results of this study.

External Validity: We discuss the external validity regarding generalizability, which is to what extent findings can be generalized to and across populations of persons, settings, and time [27]. Although the results of assessing Agility are specific to the participated teams, the conclusions are not context specific. Therefore, we believe that a similar study on different teams could result in nominating the same tool for assessing/profiling Agility.

There is not much reason to believe that the results can be generalized over time because the studied may evolve and new tools will be introduced to support Agility assessment.

Finally, the differences in the results achieved by S1, S2, and S3 might indicate that conducting the same research in different context might result in finding other tools than S3 as the best representative of the Agile status.

5.6. Conclusions

In this research, we studied different commercial tools for assessing Agility and further examined three of them in an industrial case study with two software development teams.

In response to RQ1, a summary was provided in Section 5.3.1.2 and partially in Section 5.2.2. Although the initial searches indicated that several tools exist for the purpose of evaluating Agile software development, they have different focuses and perform different analyses. We studied a selection of tools based on a search on the web, and chose three of them based on the criteria described for further examination in an industrial case study.
RQ2 was answered in Section 5.5 (in particular Sections 5.5.1, 5.5.2, and 5.5.3). Although most available tools examine only the existence/absence of the Agile practices rather than the goodness of them, the participants and the researchers agreed that S3 is the most applicable tool in the studied context.

The answer to RQ3 is that different tools do not assess the Agility similarly and hence they are not expected to give the same scores to a specific team/organization. This is confirmed by the results of this case study, and we have discussed the differences in the studied context in Section 5.4. In summary, it means that Agile is not necessarily assessed in the same way and hence Agility becomes context-dependent. Each user of Agile methods or practices must decide what Agile means to them, and hence select the best assessment tool in relation to the chosen definition of Agile.

Besides the need for assessing/profiling Agile on one hand, it is not clear how it has to be evaluated. If Agility is not properly quantified, it could hinder flexibility that is the basic principle of Agile. On the other hand, rapid changes in the market, customer demands, and the technology push software organizations towards more Agility, which means if Agile is not applied properly, it could introduce chaos in the organization instead of flexibility. Therefore, some sorts of measuring Agility would be helpful.

We would like to recommend open discussions of the evaluation results with all team members and lead managers in order to prioritize the practices that are critical for the organization (e.g. are in alignment with the organizational goals). This implies a selective approach in adopting/improving Agility rather than encouraging being perfectly Agile.

Validating S3 in a separate case study on a different team is an item for future research directions. In addition, we would like to examine possible approaches of combining organizational goals and Agile assessment tool.

Acknowledgements

This work was partly funded by the Industrial Excellence Center EASE - Embedded Applications Software Engineering, (http://ease.cs.lth.se).

References


### Appendix 5.1. A sample of survey questions

<table>
<thead>
<tr>
<th>Survey 1 - Question</th>
<th>Answer</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>How is work assigned?</td>
<td>Sure</td>
<td></td>
</tr>
<tr>
<td>a) People are given specific tasks to perform (coding, analysis, etc.) by leads / managers.</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>b) People choose what they are going to work on from a backlog.</td>
<td>☒️</td>
<td></td>
</tr>
<tr>
<td>Which of the following most closely describes the ratio of business analysts to developers within your organization?</td>
<td>More sure than unsure</td>
<td></td>
</tr>
<tr>
<td>a) 1 business analyst to 4 or fewer developers.</td>
<td>☒️</td>
<td></td>
</tr>
<tr>
<td>b) 1 business analyst to 7 or fewer developers.</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>c) 1 business analyst to 8 or more developers.</td>
<td>☒️</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Survey 2 - Question</th>
<th>Answer</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which of the following most closely resembles your build process?</td>
<td>Sure</td>
<td></td>
</tr>
<tr>
<td>a) When we want to test the application, somebody builds it manually.</td>
<td>☒️</td>
<td></td>
</tr>
<tr>
<td>b) We have an automated build process that is run once a day, or less frequently.</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>c) The application is built automatically every time somebody commits a change.</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>d) Changes are only committed after developers have run a successful, fully automated pre-commit build.</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>Do you do continuous integration?</td>
<td>Unsure</td>
<td></td>
</tr>
<tr>
<td>a) No, we have no automated tests, or we don't run them regularly.</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>b) No, but we do have an automated process that builds the application and runs unit tests regularly. Periodically an attempt is made to make the tests pass.</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>c) Yes, we have an automated process that builds the application and run unit tests every time somebody checks in. Whenever the tests break, they are immediately fixed.</td>
<td>☑️</td>
<td></td>
</tr>
<tr>
<td>d) Yes, we have a pipelined build and deployment process that builds, runs unit tests, and then subjects the generated artifacts to automated functional and integration testing.</td>
<td>☑️</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Survey 3 - Question</th>
<th>Answer</th>
<th>Sure</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team members are kept together as long as possible.</td>
<td>True</td>
<td>Sure</td>
<td></td>
</tr>
<tr>
<td>Testers and programmers are on the same team.</td>
<td>True</td>
<td>Sure</td>
<td></td>
</tr>
<tr>
<td>Teams have 5-9 people on them.</td>
<td>True</td>
<td>Sure</td>
<td></td>
</tr>
<tr>
<td>Teams can determine who is on or off the team.</td>
<td>False</td>
<td>Unsure</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Survey 4 - Question</th>
<th>Answer</th>
<th>Sure</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>To what extent does your enterprise practice and measure infrastructure and operational management?</td>
<td>Not applicable</td>
<td>Sure</td>
<td></td>
</tr>
</tbody>
</table>
Appendix 5.2. A sample of interview questions

**Teamwork**
How is the team built? Composed? Located? Who does it?
How do the team work together?
Who decides on priorities or changes them?
Standup meetings? Duration?
...

**Requirements**
Who is product owner and how collaborates with the team during an iteration?
How is the requirement handling and agreement process?
...

**Planning**
When and who does the technical design?
How often the team updates the iteration burn down charts?
...

**Technical Practices**
...

**Quality**
Is unit testing done before checking in the code?
What type of testing is performed for iteration?
...

**Culture**
Is productivity on the focus or the overwork?
...

**Knowledge Creating**
How good is the team’s knowledge to Agile?
Who is present in retrospective meetings?
...
Chapter 6

Agile Practices in Global Software Engineering: Snowballing Search Method

6.1. Summary

This report summarizes the undertaken steps in the study with the purpose of systematically reviewing the current research literature on Agile methods and Global Software Engineering (GSE). When searching for relevant papers, we followed the guidelines provided by Webster and Watson [1] known as snowballing. The results were limited to peer-reviewed conference papers or journal articles published between 1999 and 2009.

The primary purpose for conducting this study was to compare its results with the results of an already accomplished literature review in the same area which is presented in Chapter 3. Hence, the synthesis was made similarly through classifying the papers into different categories (e.g. publication year, contribution type, research method).

At the end 61 papers were judged as primary for further analysis. More details on the research methodology and the results are given in the following sections.

6.2. Motivations and Objectives

This research was planned to perform a systematic review that covers most common Agile methods in GSE settings with the same goals as another study presented in Chapter 3 (we refer to it as SLR in the remaining). However, the major objective was to use the results of this study in our next research, which will compare the findings of this study against the findings of SLR.

Therefore, for the purpose of comparisons, the data analysis was done as similar as possible to SLR study. The details on data analysis can be accessed in Chapter 3.

6.3. Research Method and Conduct

The study was designed to perform a systematic review of the existing research literature in the area of Agile in combination with GSE. We have followed the guidelines provided by Webster and Watson [1] (snowballing) in conducting the searches for finding relevant papers with a minor modification, which will be elaborated later.

6.3.1. Research Questions

The research questions were the same as in SLR study and presented as follows.

RQ1: What is reported in the peer-reviewed research literature about Agile practices in GSE?

In order to answer this question, the current research literature was explored and summarized through conducting a systematic literature review study utilizing snowballing search instrument.
RQ2: Which Agile practices in which GSE settings, under which circumstances have been successfully applied?

To answer this question, the same as in SLR study, the results of the systematic review were synthesized and successful empirical cases reported in the literature were analyzed carefully.

6.3.2. Search Strategy

This study was performed during 2010-2011 with exactly the same purpose and the same research questions as in SLR. The difference between the two studies was the way that the relevant papers and articles were extracted from the published research. The time between the two searches was a couple of months and the time between the syntheses was around eight months, and hence the details about specific papers found in the first search were not fresh in the mind of the researcher. Thus making the searches reasonably independent.

This study utilizes the snowballing search approach [1] and runs searches only in Google Scholar (http://scholar.google.com) instead of formulating search strings for each database separately as performed in SLR. The search in Google Scholar provides a starting set of papers to conduct the snowballing search.

The initial search was done only for the publication year 2009. Based on the search results, the keywords were refined if required, and searches were re-conducted. The summary of the process is shown in Figure 6.1.

The search in Google Scholar was limited to publication year 2009, but the keywords could change based on the findings. All papers were judged based on the abstract and if it was relevant, its list of references was checked for finding more relevant papers. The process stopped when we could not add more relevant papers.

In order to make the studies as comparable as possible, we kept the search terms and keywords as similar as possible to the SLR study and also applied the same constraints on searches. In addition, the same researchers were responsible for finding, evaluating, and analyzing the relevant papers in both studies in order to minimize the diversity in data collection and data analysis.

Figure 6.1. Search strategy and process
6.3.3. **Snowballing**

The snowballing search method [1] can be summarized in three steps: 1) starting the searches in the leading journals and/or the conference proceedings. 2) going backward by reviewing the reference lists of the relevant articles found in step 1. 3) going forward by identifying articles citing the key articles identified in the previous steps.

6.3.4. **Data Retrieval**

The search area was limited to the title of the papers due to limitations of the Google Scholar. Otherwise, the number of hits could increase hundred times. The search string can be summarized as: (Agile OR scrum OR lean OR “extreme programming” OR “pair programming” OR XP) AND (distributed OR global OR virtual OR GSE OR GSD OR DSD OR dispersed OR offshore OR outsource OR spread OR offshoring OR outsourcing).

The other limitations were on the language to be English, the publication year to be 2009, only within Engineering, Computer Science, and Mathematics, and it had to be at least summaries (citations were excluded), and only articles and patents.

We tried to make this search as similar as possible to the searches in the SLR study, and therefore similar limitations were applied.

6.3.5. **Inclusion Process**

This process was also the same as in SLR. We found 38 papers in the initial search. The decision on inclusion/exclusion was made based only on the abstract since full-text was not available for all papers, and in addition it is time consuming to evaluate the full text of all papers. Based on the evidence found in the title, abstract or keywords implicitly or explicitly, the papers were categorized as “relevant”, “irrelevant” or “maybe relevant”. The steps taken to extract the final set of studies for further synthesis are summarized in Figure 6.2.

Then, the list of “irrelevant” and “maybe relevant” papers was given to the second researcher without showing the previous judgments. Having both judgments, it was decided not to include papers with one “irrelevant” vote and one “maybe relevant” and to include papers with two “maybe relevant” votes in the further analysis.

![Figure 6.2. Inclusion process and results](image-url)
If the full paper was not accessible, an email sent to the main or second author asking for the paper in pdf. In addition, papers with no result or the same content as other studies were excluded. Finally, 61 studies were finally selected as primary papers for data extraction and synthesis.

6.3.6. Data Extraction and Synthesis

The data extraction was designed to be the same as SLR study because the comparison was feasible only if we made similar analysis on the data. Hence, MS Excel was used for data extraction and collection. We also used the same classification scheme as in SLR for categorizing the research type for papers. All 61 papers were fully read, data synthesis was made, and the required items were extracted, coded, and stored in Excel. Finally, several descriptive classifications of the content of the studied papers were made with respect to research methodology, empirical background, findings, participants, and context of the studies.

6.4. Results

We studied the full-text of all included papers to extract required information for our analyses. This section presents the results of analyzing the extracted data.

6.4.1. Results of Literature Review

The outcome of the selection phase was 61 peer-reviewed papers and articles. Table 6.1 shows the distinctive number of papers for each year (1999-2009). The maximum was in 2009 with 13 papers, and no relevant paper was found in 1999 as well as few papers in 2000, 2001, and 2002. Although the number of papers has increased in 2003, and 2004, it has dropped down in 2005. The greater number of papers in 2006 to 2009 indicates that the interest for combining GSE and Agile has grown in the last few years.

Table 6.1. Distribution of papers over the studied years

<table>
<thead>
<tr>
<th>Year</th>
<th>1999</th>
<th>2000</th>
<th>2001</th>
<th>2002</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of papers</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>10</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>

The same classification scheme as was used in the SLR was used for categorizing the papers according to their research type. The results of the classification are presented in Figure 6.3. The majority of the current research is in the form of experience reports, in which practitioners have reported their own experiences on a particular issue and the method used to mitigate it. The distribution of different research types over studied years pinpoints the need for conducting more philosophical, validation, and evaluation research.
Figure 6.4 presents the mapping of Agile practices with distribution settings. “Agile” as a general term together with “distributed team” seems to be the most used combination. It indicates that the contextual information in the current literature is not sufficiently documented. Hence, the experience reports might not be of much use for others.

The following section elaborates more on the successful cases in the available research. The failure stories were excluded in alignment with the research questions.

6.4.2. Successful Applications

We found 61 empirical studies in total and 42 success stories among them. If N projects were discussed in the paper, we counted the success/failure number for each of the projects as 1/N. For example, if a paper reports two projects, one successful and one failure, we added 0.5 to the successful cases and 0.5 to the failure ones.

The most applied combination of Agile methods and distribution settings are Agile- distributed team, Extreme Programming (XP)-distributed team, and Scrum-distributed team. In the majority of the studies papers, the applied Agile method is addressed as “Agile” and distribution setting is mentioned as “distribution team” without any details, which highlights the incompleteness of the contextual and background information in the current literature.

6.4.2.1. Countries Involved in Agile GSE

The countries involved in Agile GSE are summarized in Table 6.2. Countries represented as customer are the main sites/offices with major responsibilities in offshore developments, or the customers in outsourcing business relationships. If N countries were involved in a single relationship, the participation number for each was considered as 1/N. If M projects were reported, the participation rate for each project was divided by M.

The collaboration between USA-India is reported the most in the literature, and then distributed development within USA is also popular. There are not many Asian countries among the customers while some Asian countries like India are popular destinations for outsourcing due to availability of low-cost workforce.
6.4.2.2. Successful Agile Practices

The empirical studies that reported successful cases were explored in order to identify the applied Agile practices. The reported practices and their frequencies are summarized in Figure 6.5. According to the current literature, the “sprint/iterations”, and “standup meetings” are the activities that are efficiently practiced the most. The frequency of 13 for “sprints” indicates that 13 cases out of 42 reported successful application of this practice.

It should be noted that contextual information for the studied cases was not complete and, hence, it was difficult to extract the exact form of collaboration or task distribution among remote sites. The same discussion is valid for the cases in which “Agile” was reported as applied Agile method.
Figure 6.5. Agile practices and their frequencies in the studied papers

6.4.2.3. Research Methods

The fact that the majority of the current research is in the form of experience report (shown in Figure 6.3) was confirmed when categorizing the papers (see Figure 6.6). Most experience reports and opinion papers were categorized as qualitative or unclear, and the research method was identified to be either unclear or a case study. All terminologies and definitions are similar to the SLR.

Figure 6.6. Research method classifications for the studied papers
The majority of the successful cases were qualitative studies in which either a case study was reported/analyzed. The number of reported experiments in quantitative studies is also high. Too few studies used both qualitative and quantitative approaches while the research method could not be identified for 20% of the studies.

6.4.2.4. Contributions and Means of Analysis

Figure 6.7 presents the contributions and means of analysis for the studied papers. As it was expected from previous analyses on research types and methods, the majority represents problem reports and lessons learned as their contribution. Some studies implemented tools to be used in distributed Agile.

![Figure 6.7. Contribution and means of analysis of the papers](image)

6.4.2.5. Context elements for each Agile practice

For each extracted practice, we have visualized the information regarding in which specific setting it has been efficiently applied. It is presented in Figure 6.8 to Figure 6.13.

These figures represent the combination of GSE and Agile methods in which each practice is most efficient according to the available research literature. For example, Figure 6.8 shows that “spring planning” is mostly used when distributed teams practiced XP method.

It also indicates that “sprint/iterations” is most common when combining Scrum practices and offshore development, “continuous integration” is efficient in the combination of XP and offshore, and “standup meetings” is practiced efficiently when distributed teams applied Scrum method.
Figure 6.8. Context for “standup meetings”, “sprint/iterations”, “continuous integration”,
“sprint planning”

Figure 6.9. Context for “pair programming”, “retrospectives”, “test driven development”,
“sprint review”
Figure 6.10. Context for “onsite customer”, “scrum of scrums”, “unit/integrated testing”, “backlog”

Figure 6.11. Context for “refactoring”, “coding standards”, “continuous build”, “planning game”
Figure 6.12. Context for “collective code ownership”, “scrum master”, “release planning meeting”, “simple design”

Figure 6.13. Context for “feature driven development”, “40-hours week”, “one team”, “maintenance sprint”
6.4.2.6. Details of Successful Cases

In data analysis, we synthesized which Agile method was combined with which distribution setting, which practices were successfully applied for that combination, what countries were involved, and what were the main characteristics of the project e.g. size, domain, duration (see Appendix 6.2 for details). In the following, a summary is presented for each combination of Agile and GSE according to the reported information in the studied papers.

In the studies with sufficient contextual information, most cases reported the team that was globally distributed and worked for a long time period on a small to medium size project. All assumptions for data extraction are the same as in SLR.

The most efficient combination of Agile methods and GSE settings are found to be XP-distributed team and Scrum-distributed team each with 7 identical papers. In the following, Agile practices are extracted from the studied papers using their wordings.

**Extreme Programming – Offshore**: In XP-Offshore combination, USA-India collaboration seemed to be the most popular, and “continuous integration” and “sprint/iterations” is reported as the most efficient practice.

**Extreme Programming – Outsource**: A lot of single practices are reported for this combination, and Japan-Vietnam collaboration is reported the most.

**Extreme Programming – Distributed team**: Using “test driven development” was the most effective practice in XP-distributed team setting, and USA seemed to be the owner of most projects.

**Extreme Programming – Virtual team**: Only one paper was found that addressed XP and Virtual team, and too few practices such as “standup meetings”, “automated testing”, “Pair Programming”, “onsite/proxy customer”, and “enough documentation” were reported. However, countries involved were not clearly specified.

**Extreme Programming – Open source**: We found one paper that addressed XP and open source, but not enough information was provided on the context.

**Scrum – Offshore**: In most cases, USA had offshored within the country. The most reported practice was “sprint/iterations”.

**Scrum – Outsource**: The outsourcing company was mostly located in USA, and “one team/sit together” was reported as the most successful practices.

**Scrum – Distributed team**: USA is involved the most, and the most efficient practice was “backlog”.

**Agile – Offshore**: Here, USA is the most offshoring initiator, and the efficient practice is “sprint planning”.

**Agile – Outsource**: Only one study was found that reported collaboration among UK, Romania, and India with a few practices.

**Agile – Distributed team**: In this setting, Finland and India were the most popular countries, and “standup meetings” and “sprint/iterations” were the most popular practices.

6.4.3. Limitations

Since all examined databases in the SLR study are covered in the Google Scholar, the limitations of this research are the same as the SLR study. Detailed discussions can be accessed in Chapter 3 and Chapter 5.

References

Appendix 6.1. List of included papers


Appendix 6.2. Mapping practices and distributions

The assumptions and the synthesis made are the same as in Appendix 3.3 in Chapter 3.

<table>
<thead>
<tr>
<th>A. XP – Offshore [P4][P38][P39][P40]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Countries:</strong> USA: 1.58, India: 0.75, UK: 0.33, Singapore: 0.33</td>
<td></td>
</tr>
<tr>
<td><strong>Project domain:</strong> Telecom: 2, Unclear: 1.5</td>
<td></td>
</tr>
<tr>
<td><strong>Project duration:</strong> Long: 1, Medium: 1, Unclear: 2.5</td>
<td></td>
</tr>
<tr>
<td><strong>Project size:</strong> Small: 1, Unclear: 2.5</td>
<td></td>
</tr>
<tr>
<td><strong>Knowledge area:</strong> Requirements: 0.5, Design: 0.95, Construction: 0.83, Testing: 0.95, Maintenance: 0.12</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B. XP – Outsource [P12][P36]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Countries:</strong> USA: 0.12, New Zealand: 0.12, Japan: 0.5, Vietnam: 0.5</td>
<td></td>
</tr>
<tr>
<td><strong>Project domain:</strong> Web application: 1, Unclear: 0.5</td>
<td></td>
</tr>
<tr>
<td><strong>Project duration:</strong> Short: 1, Medium: 0.5</td>
<td></td>
</tr>
<tr>
<td><strong>Project size:</strong> Small: 1.5</td>
<td></td>
</tr>
<tr>
<td><strong>Knowledge area:</strong> Requirement: 0.37, Design: 0.37, Construction: 0.37, Testing: 0.37</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C. XP – Distributed team [P6][P7][P11][P24][P26][P27][P33][P34][P35]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Countries:</strong> USA: 1.74, Ireland: 0.25, Denmark: 0.25, Australia: 0.25, Malaysia: 0.25, UK: 0.16, India: 0.41, Unclear: 2, Germany: 0.25, Italy: 0.58, Czech Republic: 0.5, Brazil: 0.33</td>
<td></td>
</tr>
<tr>
<td><strong>Project domain:</strong> Telecom: 1, Web: 1, Unclear: 3, App: 1.5, Commercial: 1</td>
<td></td>
</tr>
<tr>
<td><strong>Project duration:</strong> Long: 0.5, Medium: 1, Unclear: 5.5</td>
<td></td>
</tr>
<tr>
<td><strong>Project size:</strong> Small: 0.5, Large: 0.5, Unclear: 5.5</td>
<td></td>
</tr>
<tr>
<td><strong>Knowledge area:</strong> Testing: 0.12, Construction: 0.35, Design: 0.35, Requirement: 0.35, SE Management: 1, Unclear: 4.5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D. XP – Virtual team [P16]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Countries:</strong> Unclear: 1</td>
<td></td>
</tr>
<tr>
<td><strong>Project domain:</strong> Unclear: 1</td>
<td></td>
</tr>
<tr>
<td><strong>Project duration:</strong> Long: 1</td>
<td></td>
</tr>
<tr>
<td><strong>Project size:</strong> Short: 1</td>
<td></td>
</tr>
<tr>
<td><strong>Knowledge area:</strong> Requirement: 0.25, Design: 0.25, Construction: 0.25, Testing: 0.25</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E. XP – Open Source [P41]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Countries:</strong> Unclear: 1</td>
<td></td>
</tr>
<tr>
<td><strong>Project domain:</strong> Unclear: 1</td>
<td></td>
</tr>
<tr>
<td><strong>Project duration:</strong> Unclear: 1</td>
<td></td>
</tr>
<tr>
<td><strong>Project size:</strong> Large: 1</td>
<td></td>
</tr>
<tr>
<td><strong>Knowledge area:</strong> Unclear: 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>F. Scrum – Offshore [P3][P4][P10][P14][P25]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Countries:</strong> Canada: 0.5, Russia: 0.66, USA: 1.75, Finland: 0.27, Latvia: 0.11, Germany: 0.11, Norway: 0.16, Malaysia: 0.16, India: 0.25</td>
<td></td>
</tr>
<tr>
<td><strong>Project domain:</strong> Web Application: 1.5, Logistics: 1, Application: 1, Unclear: 0.5</td>
<td></td>
</tr>
<tr>
<td><strong>Project duration:</strong> Unclear: 3, Long: 1</td>
<td></td>
</tr>
<tr>
<td><strong>Project size:</strong> Unclear: 2, Small: 2</td>
<td></td>
</tr>
<tr>
<td><strong>Knowledge area:</strong> Construction: 0.87, Testing: 0.87, Design: 0.12, Maintenance: 0.12</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>G. Scrum – Outsource [P10][P20][P21][P22][P37]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Countries:</strong> USA: 1.5, Russia: 0.5, India: 1.5, Netherlands: 1</td>
<td></td>
</tr>
<tr>
<td><strong>Project domain:</strong> Web Application: 1.5, Unclear: 3</td>
<td></td>
</tr>
<tr>
<td><strong>Project duration:</strong> Long: 3, Unclear: 1.5</td>
<td></td>
</tr>
<tr>
<td><strong>Project size:</strong> Unclear: 2.5, Medium: 2</td>
<td></td>
</tr>
<tr>
<td><strong>Knowledge area:</strong> SE Management: 1, Unclear: 1, Requirement: 0.5, Design: 0.5, Construction: 0.75, Testing: 0.75</td>
<td></td>
</tr>
</tbody>
</table>
### H. Scrum – Distributed team [P1][P5][P7][P13][P17][P26][P27][P30][P33]

| Countries: | Unclear: 1, Norway: 0.75, Malaysia: 0.75, USA: 2.49, Israel: 0.33, France: 0.33, India: 0.41 |
| Project domain: | Application: 1.5, Unclear: 5.5 |
| Project duration: | Long: 2.5, Unclear: 4.5 |
| Project size: | Small: 1.5, Large: 2.5, Unclear: 3 |
| Knowledge area: | Construction: 1.37, SE Management: 1, Unclear: 3.5, Requirement: 0.37, Design: 0.37, Testing: 0.37 |

### I. Agile – Offshore [P2][P8][P18][P23]

| Countries: | USA: 1.5, India: 1.66, UK: 0.16, Romania: 0.16 |
| Project domain: | Business critical: 1, Web Application: 1, Unclear: 1.5 |
| Project duration: | Long: 1, Unclear: 2.5 |
| Project size: | Unclear: 2, Medium: 2 |
| Knowledge area: | Maintenance: 1, Unclear: 1.5, Requirement: 0.25, Design: 0.25, Construction: 0.25, Testing: 0.25 |

### J. Agile – Outsource [P18]

| Countries: | UK: 0.16, Romania: 0.16, India: 0.16 |
| Project domain: | Unclear: 0.5 |
| Project duration: | Unclear: 0.5 |
| Project size: | Medium: 1 |
| Knowledge area: | Unclear: 0.5 |

### K. Agile – Distributed team [P9][P15][P28][P31][P41]

| Countries: | Unclear: 1, India: 1.43, USA: 1, Finland: 1.56, Australia: 0.333, UK: 0.33 |
| Project domain: | Telecom: 3, Application: 1, Unclear: 2 |
| Project duration: | Unclear: 4, Long: 1, Short: 1 |
| Project size: | Unclear: 3, Small: 2, Medium: 1 |
| Knowledge area: | Construction: 2.5, Support: 0.5, Testing: 1, Unclear: 2 |
ABSTRACT

Context: Distributed teams characterize Global Software Engineering (GSE). GSE stakeholders are from different cultures, geographic places and potentially time zones. These conditions have significant consequences on communication, coordination and control of software projects. Given these constraints, distributed teams need to highly rely on each other. Trust is the glue that holds them together and enables more open communication, which increases their performance and quality of delivered products. Simultaneously, in striving for more efficient software development approaches, Agile values and principles were formulated. Agile methods encourage establishing close collaboration between customers and developers, continuous requirements gathering and frequent face-to-face communications.

Objective: The major objective of the research is to study efficient software development approaches particularly in (globally) distributed settings. Thus, the dynamics of trust in GSE are investigated for bringing useful trust improvement suggestions to project managers. Furthermore, Agile practices that have been efficiently applied in GSE are identified through two different systematic literature review approaches (i.e. systematic literature review and backward snowballing). The differences identified in the use of Agile practices lead to a need to better understand and assess Agility.

Method: The research methods, include systematic literature reviews and case studies, are applied in different empirical cases. Then, a variety of secondary data collection methods are utilized such as semi-structured interviews, questionnaires, open discussions and presentations.

Result: Achieving trust was realized to be crucial and the success factor for trust was the “awareness” of particular GSE challenges, which shall be communicated properly to all distributed team members and proper actions shall be taken to address them. Besides, the literature indicated several successful combinations of Agile and GSE. However, despite utilizing two different literature search methods the identified patterns were similar. The most common practices were “standup meetings” and “sprints/iterations”.

Nevertheless, the current literature reports “Agile” as a general term and “distributed team” as the most common team/organization setting, which motivated examining the applicability of existing Agile assessment tools in an industrial setting. We found one of the studied tools sufficiently applicable in the context of the case organization.

Conclusions: Trust achievement is crucial for efficient GSE collaborations regardless of the applied software development approach. Although Agile promotes trust among team members, it was formulated without considering teams’ distribution. Hence, combining Agile and GSE is challenging.

The literature contains several successful cases of implementing Agile in GSE while practitioners and researchers are not yet consistent regarding their perception of Agile practices and documenting them. Therefore, they need to collaborate closely, illustrate the practices, agree on the terminology, how to document the context, and how to profile/assess Agility. For this purpose, we examined the applicability of a set of Agile assessment tools and proposed one tool for the case organization.