



Electronic Research Archive of Blekinge Institute of Technology  
<http://www.bth.se/fou/>

This is an author produced version of a journal paper. The paper has been peer-reviewed but may not include the final publisher proof-corrections or journal pagination.

Citation for the published Journal paper:

Title:

Author:

Journal:

Year:

Vol.

Issue:

Pagination:

URL/DOI to the paper:

Access to the published version may require subscription.

Published with permission from:

# Global Software Engineering and Agile Practices: A Systematic Review

Samireh Jalali and Claes Wohlin

Blekinge Institute of Technology, School of Computing, SE- 371 79 Karlskrona, Sweden

## ABSTRACT

Agile practices have received attention from industry as an alternative to plan-driven software development approaches. Agile encourages e.g. small self-organized collocated 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 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

## 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 [27].

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 collocated teams [20].

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 collocated 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. [26][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 [17]. 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 2 gives a brief background and summarizes related work. Section 3 discusses the research methodology and explains different steps of conducting this systematic review. The results of the study are presented in Section 4, and discussion and observations around them are provided in Section 5. Finally, conclusions and future research directions are presented in Section 6.

## 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.

### 2.1. Agile Practices

Agile methods consist of a set of practices for software development that have been created by experienced practitioners [28], 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" [24].

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].

### 2.2. Global Software Engineering

Geographically distributed software development teams characterize distributed software development, whilst globally distributed teams characterize global software development [19]. In this study, we have considered both as GSE. The description of different terms related to GSE is inspired by [19], 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.

### 2.3. Agile Practices in Global Software Engineering

Although Agile methods are well suited when customers and developers are collocated 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. [26][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.

#### 2.4. *Related Work*

Here, a summary of the previous relevant research is presented. Systematic review studies on Agile methods or/and 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 Dingsøy [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 Smite et al. [22] 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. [23] conducted a study in 2006 to evaluate the usefulness for practitioners of the existing research on Agile global software development. 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. [16] 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.

#### 2.5. *Motivations and Objectives*

Confirming the findings of the previous works [22][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 [22]. 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. RESEARCH METHOD AND CONDUCT

The research was designed to be a systematic literature review following the guidelines provided by Kitchenham and Charters [15]. 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. [17]. This paper presents all steps taken in designing and conducting the systematic review, and the results.

#### 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:

*RQ.1* 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.

*RQ.2* 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.

#### 3.2. *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 1.

#### 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>) and Scopus (<http://www.scopus.com>).

#### 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.

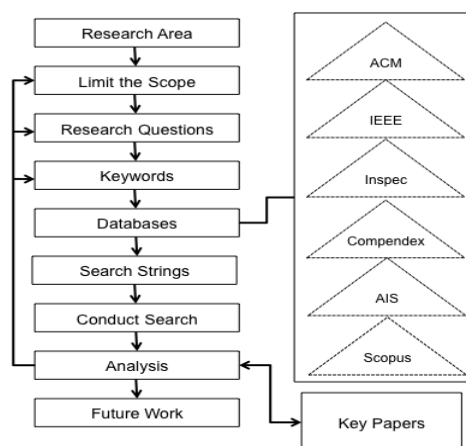


Figure 1. Search strategy and process

*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.

### 3.5. Inclusion Process

The steps taken to extract the final set of studies for further synthesis are summarized in Figure 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.



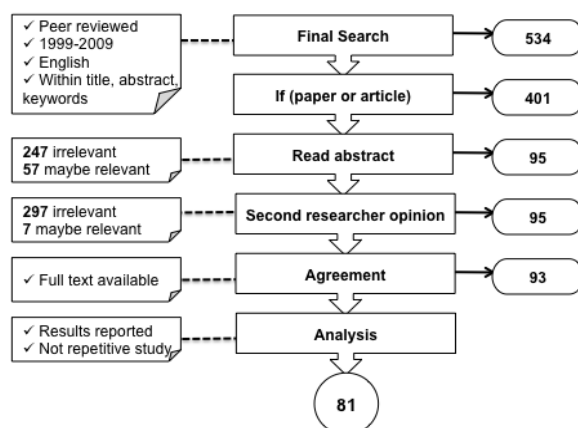


Figure 2. Inclusion process and results

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.

### 3.6. Data Extraction and Synthesis

The guidelines provided by Petersen et al. [17] 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 1 in [14]). 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. [25] 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.

#### 4. RESULTS

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

##### 4.1. Results of Literature Review

The outcome of the selection phase was 81 peer-reviewed papers and articles. TABLE I. 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 I. DISTRIBUTION OF PAPERS OVER THE STUDIED YEARS

	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
ACM						2	2	3			2
IEEE					1	2	1	2	6	15	9
Compendex				1		1	2	4	4	2	2
Inspec					1	5	1	3	2	1	
AIS									1	2	
Scopus									1		4
<b>Total</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>10</b>	<b>6</b>	<b>12</b>	<b>14</b>	<b>20</b>	<b>17</b>

The classification scheme explained in Section 3.6 was used for classifying the papers based on the research type. The results of the categorization are presented in Figure 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 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 [21].

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.

##### 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.



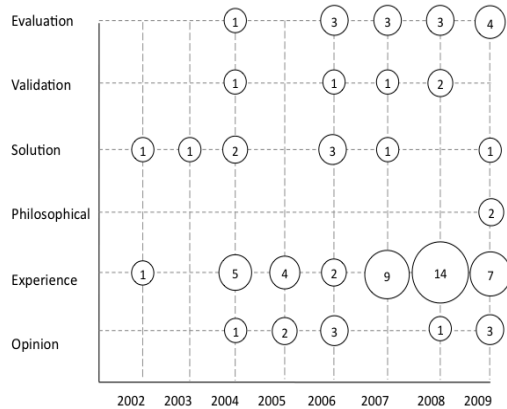


Figure 3. Distribution of research types over the studied years

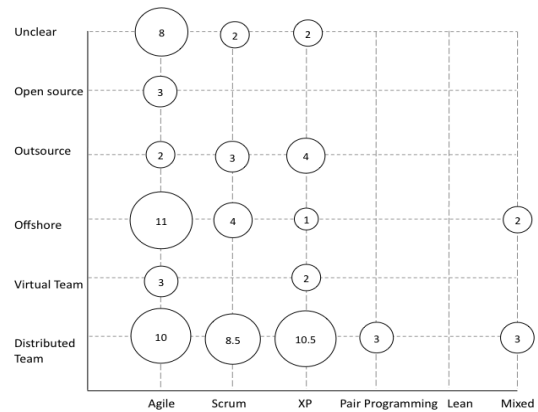


Figure 4. Mapping Agile practices and distribution types

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.

#### 4.2.1. Countries Involved in Agile GSE

The countries involved in Agile GSE are summarized in TABLE II. 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.

The collaborations between USA and India are reported the most in literature. According to TABLE II, 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.

#### 4.2.2. Research Methods

As Figure 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 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 [8] on research design approaches.

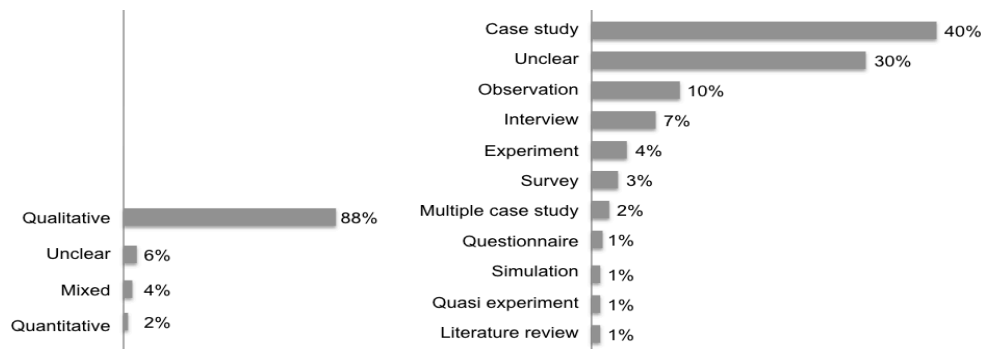


Figure 5. Research Method classifications for the studied papers

TABLE II. COUNTRIES INVOLVED IN GSE

Countries involved in Agile GSE		Customer												
		USA	Finland	Australia	Norway	China	UK	Denmark	Netherland	Germany	Canada	New Zealand	Unclear	
Supplier	India	10.5				1	1		1	1				
	USA	3					0.5				1			
	Ireland	2												
	Philippines	1												
	Ukraine	1												
	Poland	1												
	China	1												
	Canada	0.5												
	France	0.5												
	Italy	0.5												
	Brazil	0.5												
	Czech Republic	0.5												
	Russia	0.5						1		0.2	1			
	Israel	0.5												
	Norway	0.5												
	Finland		1							0.2				
	Romania						0.5							
	Latvia		0.3											
	Malaysia			1	1.3									
	Hong Kong					0.3								
Unclear	3.2										0.5	14		

As Figure 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.

#### 4.2.3. Contributions and Means of Analysis

Figure 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 collocated development.

#### 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 in [14] 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.

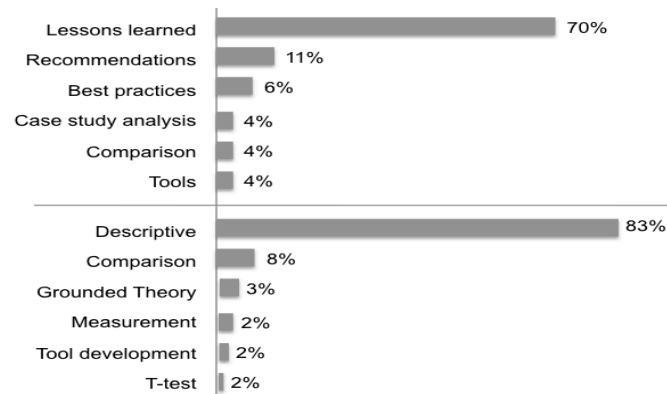


Figure 6. Contribution and means of analysis of the papers

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  $\leq$  20-person < Medium  $\leq$  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 in [14].

*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 “stand up 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 “stand up 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 “stand up 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 “stand up

meetings” and “sprint/iterations” were the most popular practices.

*Agile – Open source*: “stand up 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.

#### 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 7. The frequencies represent the According to the available research literature, “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.

#### 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.

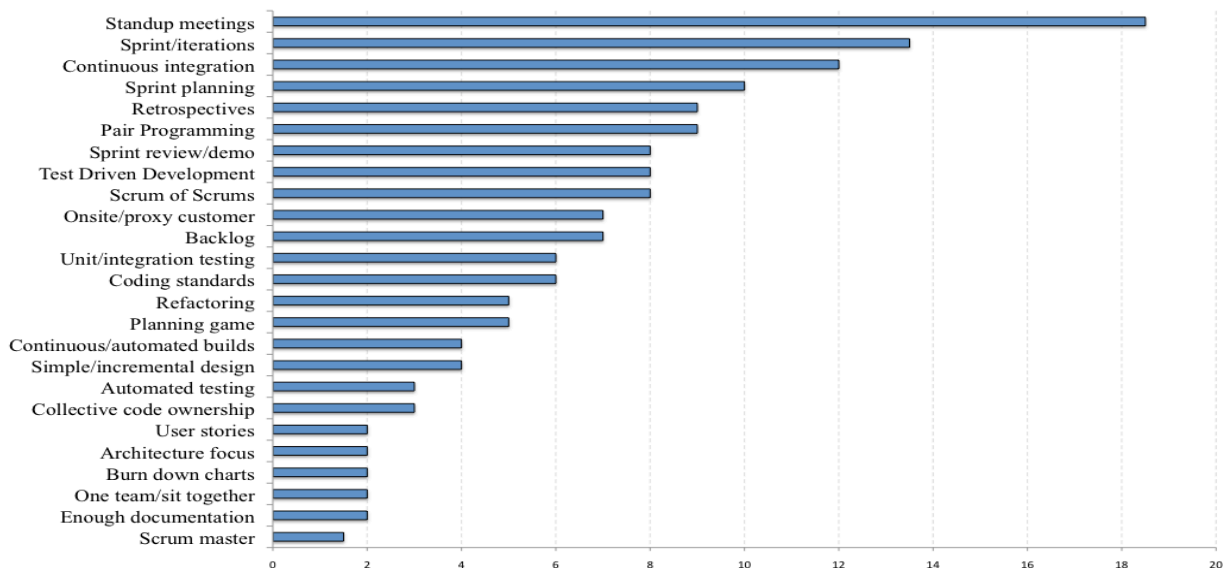


Figure 7. Agile practices and their frequencies in the studied papers

- |                                |  |
|--------------------------------|--|
| 1) XP – Distributed team: 9    | 9) Agile – Open source: 2                  |
| 2) Agile – Offshore: 7.5       | 10) Agile – Outsource: 1.5                 |
| 3) Scrum – Distributed team: 7 | 11) Agile – Virtual team: 1                |
| 4) Scrum – Offshore: 6.5       | 12) XP – Unclear: 1                        |
| 5) Agile – Distributed team: 6 | 13) XP – Virtual team: 1                   |
| 6) Scrum – Outsource: 4.5      | 14) Pair Programming – Distributed team: 1 |
| 7) XP – Offshore: 3            |  |
| 8) XP – Outsource: 3           |  |

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.

#### 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. 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 catalogs. 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.

## 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 [S56 in 14], and in some others, a distributed company decided to switch to Agile e.g. due to failure of a process-driven development approach [S22 in 14]. 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 from 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. S74, S22 in [14]). 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. S24, S71 in [14]). 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 (S16, S42 in [14]), whilst others believe they are contradictory in nature and it emphasizes the GSE challenges [S9 in 14]. 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 [18] 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 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 (S40, S41 in [14]), the other contextual project factors are not clearly reported.

Large Agile teams may face complications and issues in their communications [S31 in 14], 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. S31 in 14] 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 [S77 in 14]. 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. [S24 in 14]). In some other cases, XP and Scrum practices were selectively applied in situations that is claimed to be either XP or Scrum (e.g. [S64 in 14]).

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.

## 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 RQ.1. 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 RQ.2. 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 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 4.2.6). USA-India is reported to be the most common applied distributed business relationship with successful results (TABLE II. ).

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.

These 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.

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

## REFERENCES

1. P. Abrahamsson, J. Warsta, M.T. Siponen, J. Ronkainen, “New Directions on Agile Methods: a Comparative Analysis”, Proceedings of the 25th International Conference on Software Engineering, ACM Press, 2003, pp. 244-254.
2. A. Abran, J.W. Moore, Guide to the Software Engineering Body of Knowledge (SWEBOK®), IEEE Computer Society 2004 Guide, Angela Burgess, 2004.
3. B. Boehm, R. Turner, “Management Challenges to Implement Agile Processes in Traditional Development Organizations”, IEEE Software (22)5, 2005, pp. 30-39.
4. I. Bose, “Lessons Learned From Distributed Agile Software Projects: A Case-based Analysis”, Communications of the Association for Information Systems 23(34), 2008.
5. S. Bowen, F. Maurer, “Process Support and Knowledge Management for Virtual Teams Doing Agile Software Development”, Proceedings of the 26th Annual International Computer Software and Applications Conference, IEEE Computer Society Press, 2002, pp. 1118-1120.
6. K. Conboy, B. Fitzgerald, “Toward a conceptual framework of agile methods: a study of agility in different disciplines”, Proceedings of XP/Agile Universe, Springer Verlag, 2004.
7. G. Corbitt, L.R. Gardiner, L.K. Wright, “A Comparison of Team Developmental Stages, Trust and Performance for Virtual versus Face-to-Face Teams”, Proceedings of the 37th Hawaii International Conference on System Sciences, 2004.
8. J.W. Creswell, Research design: qualitative, quantitative, and mixed method approaches, Second Edition, SAGE, ISBN: 0761924426, 9780761924425, 2003.
9. A. Danait, “Agile Offshore Techniques - a Case Study”, Proceedings Agile 2005, 2005, pp. 214-17.
10. T. Dybå, T. Dingsøyr, “Empirical Studies of Agile Software Development: a Systematic Review”, Journal of Information and Software Technology 50 (2008), 2008, pp. 833-859.
11. J. Erickson, K. Lyytinen, K. Siau, “Agile Modeling, Agile Software Development, and Extreme Programming: the State of Research”, Journal of Database Management 16 (4), 2005, pp. 88-100.
12. E. Hossain, M. Babar, H. young Paik, “Using Scrum in Global Software Development: a Systematic Literature Review”, Fourth IEEE International Conference on Global Software Engineering, 2009, pp. 175-184.
13. S. Jalali, C. Wohlin, “Agile Practices in Global Software Engineering – A Systematic Map”, 5th IEEE International Conference on Global Software Engineering (ICGSE), Princeton, USA, August 2010, pp. 45-54.
14. S. Jalali, C. Wohlin, Global Software Engineering and Agile Practices: a Systematic Review, Technical report, <http://www.wohlin.eu/JalaliWohlin.pdf>, 2011.
15. B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE-2007-01, School of Computer Science and Mathematics, Keele University, 2007.
16. M. Paasivaara, S. Durasiewicz, C. Lassenius, “Using Scrum in Distributed Agile Development: A multiple Case Study”, International Conference on Global Software Engineering, 2009, pp. 195-204.
17. K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, “Systematic Mapping Studies in Software Engineering”, 12th International Conference on Evaluation and Assessment in Software Engineering, June 2008, pp. 71-80.
18. K. Petersen, C. Wohlin, “Context in Industrial Software Engineering Research”, 3rd International Symposium on Empirical Software Engineering and Measurement, 2009, pp. 401-404.
19. R. Prikladnicki, J.L.N. Audy, D. Damian, T.C. de Oliveira, “Distributed Software Development: Practices and challenges in different business strategies of offshoring and onshoring”, Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE), 2007, pp. 262-274.

Please cite as: Jalali, S. and Wohlin, C. (2011), Global software engineering and agile practices: a systematic review. *J. Softw. Maint. Evol.: Res. Pract.* doi: 10.1002/smr.561

20. H. Sharp, H. Robinson, "An Ethnographic Study of XP Practice", *Journal of Empirical Software Engineering* 9(4), 2004, pp. 353-375.
21. D. Smite, C. Wohlin, R. Feldt, T. Gorschek, "Reporting Empirical Research in Global Software Engineering: a Classification Scheme", *Proceedings of International Conference on Global Software Engineering*, 2008, pp. 173-181.
22. D. Smite, C. Wohlin, T. Gorschek, R. Feldt, "Empirical Evidence in Global Software Engineering: A Systematic Review", *Journal of Empirical Software Engineering* 15(1), 2010, pp. 91-118.
23. P.S. Taylor, D. Greer, P. Sage, G. Coleman, K. McDaid, F. Keenan, "Do Agile GSD Experience Reports Help the Practitioner?", *Proceedings of the 2006 international workshop on Global software development for the practitioner*, 2006, pp. 87-93.
24. L. Williams, A. Cockburn, "Agile software development: it's about feedback and change", *IEEE Computer* 36(6), 2003, pp. 39-43
25. R. Wieringa, N.A.M. Maiden, N.R. Mead, C. Rolland, "Requirements Engineering Paper Classification and Evaluation Criteria: a Proposal and a Discussion", *Journal of Requir. Eng.* 11(1), 2006, pp. 102-107
26. C. Young, H. Terashima, "How Did We Adapt Agile Processes to Our Distributed Development", *AGILE '08*, 2008, pp. 304-309.
27. P.J. Ågerfalk, B. Fitzgerald, "Flexible and distributed software processes: old petunias in new bowls?", *Communications of the ACM* 49(10), 2006, pp. 27-34.
28. P.J. Ågerfalk, B. Fitzgerald, H. Holmström, B. Lings, B. Lundell, E. Ó Conchúir, "A Framework for Considering Opportunities and Threats in Distributed Software Development", *Proceedings of International Workshop on Distributed Software Development*, Austrian Computer Society, 2005, pp. 47-61.

**SAMIREH JALALI** is a Ph.D. Student at Blekinge Institute of Technology, Sweden. She has a M.Sc. degree in Software Engineering from Blekinge Institute of Technology, and 4 years of industrial experience in Iran. Her main research interests are Agile software development, global software engineering, software quality, and empirical methods in software engineering. Contact her at Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden; samireh.jalali@bth.se.

**CLAES WOHLIN** is a professor of software engineering at Blekinge Institute of Technology. His research interests include empirical methods in software engineering, software quality, and process improvement. Wohlin received a PhD in communication systems from Lund University. He is the Editor-in-Chief of *Information and Software Technology* published by Elsevier. Dr. Wohlin is a member of the Royal Swedish Academy of Engineering Sciences; he is a senior member of the IEEE and a member of the ACM. Contact him at Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden; claes.wohlin@bth.se.