A systematic review of distributed Agile software engineering

A SYSTEMATIC REVIEW OF DISTRIBUTED AGILE SOFTWARE ENGINEERING

Buturab Rizvi[1], Ebrahim Bagheri[2], Dragan Gasevic[3]

[1]Athabasca University, [2]Ryerson University, [3]University of Edinburgh
corresponding author's email: bagheri@ryerson.ca

Abstract:

Context: The combination of Agile methods and distributed software development via remote teams represents an emerging approach to addressing the challenges such as late feedback, slow project timelines, and high cost, typically associated with software development projects. However, when projects are implemented using an Agile model with distributed human resources, there are a number of challenges that need to be considered and mitigated.

Objectives: The objectives of our work are multifold. First, we would like to understand the reasons and conditions that lead to the adoption of distributed agile software engineering practices. Second, we would like to investigate and find out the most important risks that threaten a distributed agile software engineering approach and what mitigation strategies exist to address them. Finally, would like to highlight which of the available approaches among the existing agile methodologies have been successfully adopted by the community. We intend to solidify our findings by exploring the strength of the evidence that has been reported in the literature.

Methods: We carried out a systematic literature review of Distributed Agile Software Engineering techniques and approaches reported from January 1, 2007 until September 31, 2012. The adopted method follows the well-established guidelines in the literature for conducting systematic literature reviews.

Results: Sixty-three distinct studies were selected and analyzed according to the inclusion and exclusion criteria, which focused on identifying only those studies from the literature that had a significant empirical or experimental aspect to them. The results revealed a significant number of scenarios reported by the industry that documented challenges and solutions in a Distributed Agile setting. Communication, Coordination, Collaboration, and Cultural issues were listed as being the areas where challenges exist.

Conclusions: This systematic literature review found time zone difference, knowledge of resources, lack of infrastructure, missing roles and responsibilities as being the primary challenges that needed to be addressed. In terms of solutions, most papers had recommended having a good infrastructure in place for communication, encouraging team members to engage in formal and informal communications, having more face-to-face visits, training human resources on DASE and organizational practices, policies, procedures, and utilizing tools to enhance the collaboration

experience. Additionally, this research provides recommendations to help improve the current state of reporting findings and results in the Distributed Agile Software Engineering domain.

Keywords: Distributed Agile Software Engineering, Empirical Results, Surveys and Field Studies, Agile, DASE, Systematic Literature Review

1. Introduction

In the last decade, research on Distributed Software Engineering (DSE) (or DSD – Distributed Software Development) has evolved rapidly. Cheaper labor, access to global talent, increase in business, faster delivery, and follow-the-sun development are just some of the many reasons why companies choose to engage in DSE. However, there are challenges that organizations face with such engagement. Challenges such as economic instability, technological, organizational, communication, team trust, and cultural issues need to be tackled by organizations and teams involved in Distributed Development. Despite the recent growth of this topic, distributed software development is still evolving. As such, the failure rate of DSE projects is higher than collocated projects [77].

DSE allows the client organization to engage in activities across one or more remote sites [77]. The combination of remote sites forms a network of sub-teams or remote teams that work together on a common goal. When DSD is implemented using Agile methodologies, such as Scrum or XP, the challenges increase. For instance, the coordination of tasks between teams becomes a more challenging endeavor for Project Managers and Leads [77].

Before DSE and outsourcing gained momentum, organizations used to outsource work to a vendor that performed single basic functions. The first documented outsourcing was in 1963 between Electronic Data Systems (EDS) and BlueCross Blue Shield of Pennsylvania where EDS was responsible for managing BlueCross' data [79]. Following the success of this initiative, EDS started to receive intake work from companies like Frito-Lay and General Motors. This is when large-scale outsourcing became noticeable. In the late 1980s, IBM signed a deal with Kodak to outsource their technology initiatives [79]. The commonality between these deals were that they were total outsourcing, where there is a transfer of the work, human resources and management, as opposed to a project, components of a project, or augmenting human resources.

As requirements became intense, software construction became more complicated. To make matters more complex, several companies were formed in the 1990s that had specific skills or had distribution rights on software [79]. This forced companies who wished to integrate or purchase such software to engage in deals with these specialized companies. This resulted in companies such as Kodak, General Motors, IBM, and EDS working with multiple vendors, which led to the introduction of DSE and the birth of several fields, such as contract management, customer relations, auditing, and benchmarking [79]. Although Gartner Group reports that 70% of

companies engage in some form of outsourcing, they estimate that a significant proportion of these companies will also have to renegotiate their contracts [78].

Agile methodologies are undertaken by organizations that are interested in delivering business value regularly and often [85]. Additionally, organizations can quickly assess the value of the product early on and decide on the fate of the project without spending too much money. The product owners focus on prioritizing the functionality, such that the core value of the project is delivered in the early releases; therefore, increasing value to the organization. As the market trend changes, the team can quickly adapt to the change.

Agile projects are known to decrease overall portfolio cost by cancelling projects and programs early on therefore emphasizing on those projects that have a good return on investment [14][23]. The core practice of Agile methodologies, such as XP, is that they prefer collocation of human resources, less documentation, and face-to-face interaction thereby eliminating communication related delays, and creating team cohesion [44]. If the above is true, then XP projects can eliminate the need for extensive and formal requirements analysis and entire design of the system early on [49]. In each iteration or Sprint, the project team will analyze, design, develop, and test only the functionality that is part of the sprint. As functionality is developed, it is integrated with the previously delivered functionality and validated. At the end of the sprint, deployment into production is scheduled resulting in quick feedback from the customers and other stakeholders.

Projects that engage in DSE are bound to face many challenges [14][23]. Even with these challenges, organizations wish to engage in DSE due to cost savings (40%), capacity of remote teams (20%), application knowledge (13%), and quick time to market (11%) [66]. [65] defines the characteristics of DSE as the following:

- Multi-sourcing – multiple distributed member involvement in a joint project; characterized by a number of collaboration partners.
- Geographic distribution – partners are located far away from each other.
- Temporal diversity – characterized by the level of working hours overlay.
- Socio-cultural diversity – level of social, ethnic and, cultural fit.
- Linguistic diversity – characterized by the level of language skills.
- Contextual diversity – level of organizational fit (diversity in process maturity and work practices).
- Political and legislative diversity - effect of cross border collaboration due to political threats or threats associated with incompatibility of laws.

Organizations that wish to engage in DSE could find that Agile methods provide the flexibility that they need [31][83]. However, the principles of DSE and Agile are not always necessarily compatible [18]. There is no collocation amongst team members resulting in no face-to-face interaction. Team cohesion does not exist due to cultural and language issues. Informal communication is also not certain depending on the time zones where sub-teams reside which result in asynchronous communication. Projects with distributed human resources could require more documentation to avoid any issues around scope misunderstanding, which is not inline with Agile characteristics [18].

In addition, XP practices such as pair programming are difficult to implement with distributed human resources. Agile revolves around informal practices and mutual adjustment where as DSE revolves around formal mechanisms and direct supervision [84]. Based on the above, it seems that Agile methods do not work with distributed software development projects. It cannot be the case that Agile practices be implemented partially as Agile insists that it be implemented entirely in order to be efficient [64]. Research of DSE and Agile related issues are ongoing and application of best practices is being piloted on several distributed projects.

With distributed human resources and Agile methodologies becoming more common, it is important to get a good understanding of the challenges faced by organizations that have implemented Distributed Agile Software Engineering (DASE) in the past [76][80]. The objective of this study is to provide an understanding of these challenges and propose solutions on ways to deal with these challenges. Results of this study will help organizations engaging in DASE by providing an overview of the distribution model used in past studies, challenges faced, and solutions implemented to deal with the challenges.

This paper is structured as follows. Section 2 provides a background discussion covering the rationale of this work, the research questions that this work intends to answer, followed by Section 3 on an overview of related work. Section 4 provides an overview of the design of this systematic literature review. Section 5 discusses the execution of the review along with threats to validity. Section 6 presents and discusses the results of the review in reference to the research questions outlined in Section 2. Section 7 highlights and discusses the main findings derived from the analysis of the results and provides a concluding discussion of the review and its findings followed by recommendations for future work.

2. Motivation and Research Questions

Systematic reviews are becoming a standard research method amongst software engineers [68]. Since its inception in 2004, systematic literature reviews have gained significant popularity among software engineers [68]. However, despite their gain in popularity, practitioners still are lacking in significant knowledge about this research method and the number of explored topics remains limited [68]. The deficiency in explored topics holds true in the area of DASE and justifies a need for more systematic literature reviews of Agile when implemented with distributed human resources.

To our knowledge, there have only been very few systematic literature reviews performed in the specific area of agile methods for global and distributed software development/engineering. The paper by Smite et al has focused on reviewing empirical evidence in global software engineering; therefore, it is not focused on aspects of agile methodologies [72]. Similarly, Jimenez and Piattini focus on some major problems within the distributed software development domain in general and provide an overview of some of the suggested solutions. The work by Hossain et al is closer to the theme of our work and focuses on the review of the role and impact of the Scrum approach on

distributed software development. However, it is only restricted to Scrum from the range of Agile methodologies. The works by Jalali and Wohlin [71][73] are the closest to our work in this paper as they have attempted to summarize the state of the art in agile practices within global software engineering until 2010 and investigate which of the agile practices have been effectively used in global software engineering. While the topic area of the work in these two papers is similar to ours, we differ in the objectives of our systematic literature reviews.

While the work by Jalali and Wohlin [71][73] focus on summarizing the state of the art and identifying the important circumstances for deploying successful agile methods for global software engineering, we focus on more fundamental research questions. The objectives of our work are multifold. First, we would like to understand the reasons and conditions that lead towards the adoption of distributed agile software engineering practices. Second, we would like to investigate and find out the most important risks that can threaten a distributed agile software engineering approach and what mitigation strategies exist to address them. Finally, would like to highlight which of the available approaches among the existing agile methodologies have been successfully adopted by the community. We intend to solidify our findings by exploring the strength of the evidence that has been reported in the literature. As we will later explain in Section 4.2, a major differentiating factor for our work compared to the earlier work by Jalali and Wohlin is that our focus has been to include only the work that have a strong empirical, experimental, or case study perspective. For this reason, our search query has been designed in such a way to include publications in DASE that have the empirical investigatory aspect to them. This is something that has not been the focus for Jalali and Wohlin.

We translate our research objectives into four specific research questions as follows:

- ***RQ1:*** *What are the conditions under which organizations choose to adopt DASE?*

  This question will help future engagers in DASE by providing a holistic picture of the circumstances that have led prior researchers and/or practitioners to adopt DASE.
- ***RQ2:*** *What are the biggest threats when adopting DASE?*

  This question will help those who wish to engage in DASE by outlining documented risks and the impact they can have on the successful delivery of DASE.
- ***RQ3:*** *What model of Agile is most adopted in DASE?*

  This question will help adopters understand which of the existing agile methodologies have had a higher success history and there is evidence from the community to show their impact.
- ***RQ4:*** *What is the strength of evidence in supporting the findings of the above questions?*

  This question will clarify the degree of strength of evidence that are available within the literature to support the findings in this systematic literature review.

In order to provide proper levels of details for abovementioned research questions, these questions are refined into several research questions. All research questions and their descriptions are recorded in Table 1.

It is important to point out that the supporting evidence to study each of these research questions is based on the information that has been reported by the community in the form of peer-reviewed publications in conferences or journals as explained later. Therefore, the research questions should be understood in that context and note should be taken when interpreting the results, as there may be work in actual practice that has not been reported in the literature and so has not been captured in our work. In light of this, the research questions should be read as, for instance, what are the conditions under which organizations choose to adopt DASE as indicated in the reported literature.

Table 1: Designated research questions for the study

#	Research Question	Description
1	What are the conditions under which organizations choose to adopt DASE?	To provide an overview of the environment in which DASE was studied.
1.1	What phase of the project lifecycle has utilized distributed human resources in Agile?	To bring forth SDLC phases that primarily used distributed human resources. The answer to this question will explain which phases of the project remote teams were engaged in.
1.2	What is the typical human resource distribution model?	To provide data on the number of teams engaged in DASE projects and how far apart these teams were (at peak).
1.3	How much experience do human resources have in Distributed Agile Software Development?	To understand the existing knowledge human resources on distributed and Agile projects.
2	What are the biggest threats when adopting DASE?	A provide details on risks and solutions
2.1	What are some of the biggest risks in DASE?	Document risks that projects have faced and issues that have risen during the course of the project.
2.2	How are risks, limitations, and mitigation strategies in DASE dealt with?	To document workarounds or mitigation strategies that projects have utilized to deal with the risks and issues, as they were uncovered.
3	What model of Agile is most adopted in DASE?	To provide an understanding on the outcome of the project.
3.1	Has one Agile model resulted in more success in distributed teams?	To understand the success rate between the different Agile models and to assess if a model stands out as being the best in DASE.
3.2	Is one Agile model shown to be worst in distributed teams?	To understand the failure rate between the different Agile models.
4	What is the strength of evidence in supporting the findings of the above questions?	To get an understanding of the overall strength of this research study.
4.1	What is the source of evidence?	To understand if research was conducted on student or employee subjects in academic or industry settings.
4.2	What is the data collection approach followed?	To understand subjects of the research, the environment it was conducted in, purpose of the study and the degree of realism.

3. Related Works

Prior to conducting this study, previous systematic literature reviews and systematic mapping studies were reviewed to ensure that the research questions defined are unique and have not been answered given same input variables. Systematic mapping and systematic literature reviews are fairly new to the field of Software Engineering and, as such, not many papers can be found. To

keep the literature review recent, an analysis was done on the scholarly papers published on this subject since 2007.

To find past literatures, search strings were formulated to combine both the distributed aspect and Agile aspect. Additionally, papers that focused on systematic literature, systematic review, or systematic map taken into consideration.

Table 2 Search terms used to find DASE literature reviews

Population	AND	Intervention
(Systematic AND (stud* OR map* or review*))	AND	(Global or Distributed) AND (Agile OR Scrum OR XP OR Pair Programming)

Search query in Table 2 was used on Google Scholar and the DBLP Computer Science Bibliography. Additionally, the query was performed on IEEExplore, ScienceDirect, SpringerLink, Wiley Online Library, and ACM.

Table 3 Summary of literature review

Ref	ID	Title	Published	Year
1	[71]	Agile Practices in Global Software Engineering - A Systematic Map	IEEE - International Conference on Global Software Engineering (ICGSE)	2010
2	[72]	Empirical evidence in global software engineering: A systematic review	ACM - Empirical Software Engineering	2009
3	[73]	Global software engineering and agile practices: A systematic review	Wiley - Journal of Software Maintenance and Evolution: Researsh and Practice	2011
4	[74]	Problems and Solutions in Distributed Software Development: A Systematic Review	SpringerLink - Software Engineering Approaches For Offshore and Outsourced Development	2009
5	[75]	Using Scrum in Global Software Development: A Systematic Literature Review	IEEE - Fourth IEEE International Conference on Global Software Engineering	2009

Table 3 lists the five resulting papers all of which were part of the literature review. Papers 1 [71], 3 [73], and 5 [75] are published papers that involve Global software engineering and Agile methods. Papers 2 [72] and 4 [74] perform review of all globally distributed software engineering projects that include various development methodologies – including Agile methods.

In paper 1, Jalali and Wohlin [71] performed a literature review to understand what is reported in the current literature about Agile methods in DSE and which Agile practices and in which DSE setting, they were successful. A total of 77 papers published between 1999 and 2009 were reviewed as part of this study. A common result of papers reviewed by Jalali and Wohlin was the documentation of issues, specific solutions, and lessons learned. Additionally, majority of the 77 papers did not document the type of distribution model or type of Agile methodology adopted. Of 60 Empirical studies, 50 projects were considered successful. Success was a result of organizations performing continuous integration, daily standup meetings, pair programming, retrospectives, scrum of scrum meetings, and TDD.

In paper 3 [73], the authors attempt to answer the same questions as in paper 1 except by creating a systematic review as opposed to paper 1 where a systematic mapping was created. This systematic review came to the conclusion that success is achieved when XP is combined with GSE.

Paper 5 [75] focused on understanding the challenges faced when scrum is used in DSE. It also helps understand what practices are used to work around the challenges faced in DSE and Scrum. A total of 20 papers published between 2003 and 2009 were reviewed. Results showed that even though Scrum has been widely adopted, it is not fully clear if Scrum can lead to successful distributed projects. Additionally, success was more common on projects where the distributed teams were within the same organization. Having said that, challenges faced in DSE when using Scrum are the same as those faced in DSE projects– such as communication, coordination, and general collaboration.

Paper 2 [72] focuses on the topic of global software engineering without focusing on a specific development methodology. The goal of this paper was to understand the state of the art in GSE and to get a feel of the strength of the empirical evidence reflected in the literature. There were 59 papers published since 2000 that were part of the review. The approach followed by this study was to understand how GSE was performed (i.e.: within an organization or by using a vendor) and understand the time-zone differences between teams. Using these factors, success and failure rates were determined. Published results state that more than half of the papers analyzed were case studies based on interviews in a controlled environment with students. The review indicated that although such research had been done for a few years and outsourcing had been practiced for up to 20 years, there was no single recipe for success. The outsourcing field in still relatively new and, as such, there is a lack of methods, techniques, and tools in an industrial context [64]. Additionally, most of the research focused on the different variables as opposed to an in-depth analysis of the various practices and techniques that would result in successful projects.

Similar to paper 2 [72], paper 4 [74] also focuses on the general topic of GSE. The flavor of this paper is to understand which processes, procedures, and strategies brought more success in GSE/DSE. Examples of processes, procedures, and strategies included CMM, CMMI, COBIT, and ITIL. An interesting point of this paper is that it discussed how procedures could impact DSE – as organizations could conduct outsourcing with companies that might have different CMM levels or could follow different frameworks (ITIL vs. PRINCE). A total of 69 papers published between 2000 and 2007 were selected as part of this research. A majority of the 69 papers were published in 2007 indicating that the area was gaining attention within in the research community. Only 25% of the reviewed papers focused on maturity models such as CMM and CMMI. Research indicated that higher maturity models resulted in added costs. This is expected since maturity models focus on processes and procedures that are not always best to strictly enforce in a distributed model.

Finally, we would like to highlight our main contributions that set us apart from the important existing systematic literature reviews that are available. As mentioned earlier, the closest systematic literature review is the works by Jalali and Wohlin [71][73]. However, these works

focus on the review of the state of the art in the area of agile methods for global software engineering and also the identification of the main approaches that have been adopted by the community. However, in our work we provided a different perspective on the literature, i.e., we first explore the underlying reasons why agile methodologies are adopted within DSD. In other words, we explore the roots and grounds for which agile methodologies were adopted. This allows for a deeper understanding of the evolution practices within DASE. Furthermore, we identify the major roadblocks and risks that hinder and threaten the successful adoption of DASE. In our opinion, this is a significant distinguishing factor for our work as it enables practitioners to understand the prospects of adopting DASE. Finally, we highlight the agile approaches that have been widely used in DASE and further solidify our findings by not only reviewing the reports of the approaches in published papers but also the strength of the evidence that is provided in support of the adopted approaches.

4. Method

This section provides the details surrounding the review protocol employed to guide the conduct of this review. It discusses the systematic review design, data source and search strategy, study selection criteria, quality assessment criteria, data extraction procedures, and data synthesis procedures.

4.1. Systematic Review Design

Based on the review protocol provided in [68], the review methods in this paper involve defining research questions, reviewing scope, conducting searches on data sources, screening papers, reviewing abstract, reviewing classification scheme, extracting data to answer research questions/properties, and documenting the results. These phases are illustrated in Figure 1.

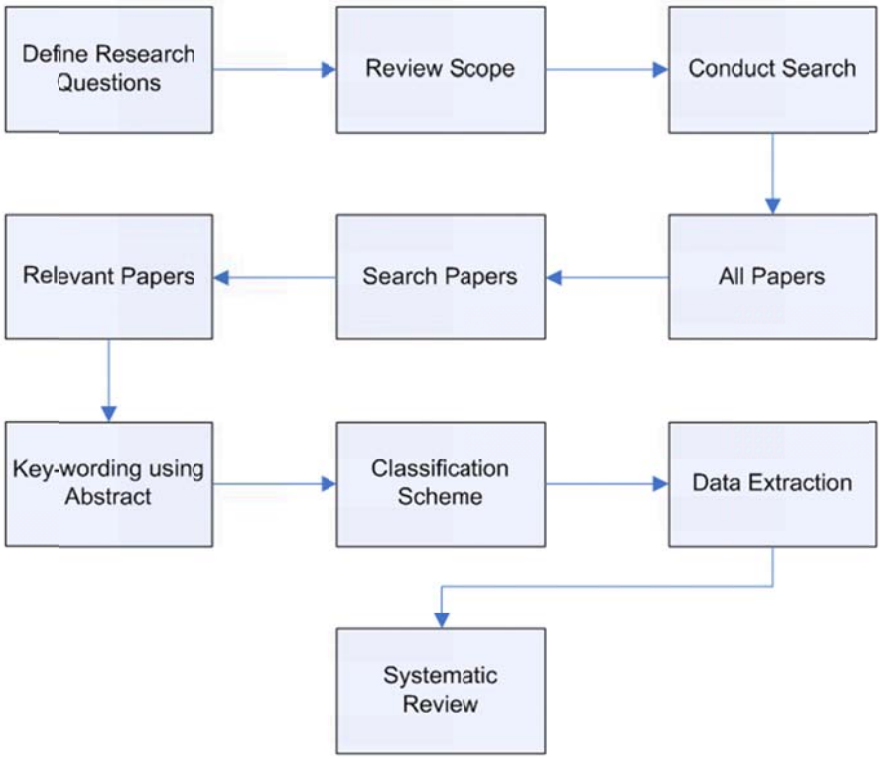

Figure 1 Systematic Literature Review Phases

In the planning phase, we developed a method using which we have conducted our review process. In this phase, we identified the portals for paper searches, the search query definition, filtering search results using inclusion and exclusion criteria, quality assessment criteria to further select appropriate papers, and data extraction process.

While conducting the review, we identified studies based on the search query, implemented our inclusion and exclusion criteria on the studies, and applied the quality-based criteria on the results. The discussion of the review execution is presented in the next chapter.

During reporting of the results, we synthesized the data extracted in the review execution phase and summarized the results. Results and analysis will be reported in Section 6. Overall, this paper follows the widely used recommendations for structuring of reports of systematic reviews outlined in Table 9 of [67].

4.2. **Data Sources and Search Strategy**

The process of identifying relevant papers in the field of Distributed Agile Software Engineering was performed on five major digital libraries, namely IEEExplore, ScienceDirect, SpringerLink, Wiley Online Library, and ACM. These portals have been consistently used in the past for performing literature reviews in Software Engineering [71][72][73][74][75].

Search criteria were set up based on the search query used in previous DASE systematic reviews [71], [72], [73], [74], and [75]. *The query was then expanded to include examples, investigations, analysis, and lessons learned.* Additionally, Pair Programming was added as an Agile model based on research conducted by [47]. The search query below was used in the data sources listed in Table 4. The query was modified for each of the data sources above such that appropriate papers were retrieved. For the conferences that had both technical and experience report tracks, no distinction was given to either type of paper as long as the papers satisfied the search terms according to Table 4.

Table 4: Search Terms used to find DASE literature

Population	**AND**	**Intervention**
(Global or Distributed) AND (Agile OR Scrum OR XP OR Pair Programming)	AND	(empiric* OR experient* OR experiment* OR experience* OR "lesson learned" OR lessons learned" OR "lesson learnt" OR "lessons learnt" OR evaluat* OR validat* OR stud* OR case* OR example* OR survey* OR investigat* OR analy*)

4.3. **Study Selection**

Once an initial pool of papers was selected snowballing technique was used to expand the list of relevant papers and then the title and abstracts of all the collected papers were reviewed. Papers that were specific to Distributed Agile software Engineering were selected. Finally, with the identification of the inclusion and the exclusion criteria, the entire paper was reviewed and compared against the criteria for further filtering. Tables 5 and 6 detail the inclusion and exclusion criteria, respectively.

Table 5: Inclusion Criteria for determining the papers for the study

Inclusion Criteria	**Rationale**
Papers where the search terms were found in the title and/or abstract.	Since the purpose of this paper is to evaluate Distributed Agile Software Engineering, the keywords should appear in the queried papers title or abstract.
Papers published between 2007 to 2012	Since the field of distributed agile is changing quickly, focusing on the last 5 years will provide relevant evaluations.
Papers where the full-text is available.	If the full-text is not available for review then there is no information to review and extract. If there is some information it is most likely unreliable.
Papers written in English.	Time constraints and language barriers restrict this review to consider papers written in English only because the author is unilingual and does not have the human resources available for translation of other languages.
Papers that are either a research paper, peer-reviewed paper, academic paper, or something of a similar nature.	Due to quality restrictions this review was limited to conducted searches in academic electronic databases. Other sources of evidence such as company journals, technical reports, and work-in-progress were avoided.
Papers that have evaluated or have used to implement a project in an Agile model using distributed human resources	Since the primary objective of this paper is to evaluate Distributed Agile Software Engineering projects, the approach of the queried paper must focus on evaluating or implementing DASE.

Table 6: Exclusion criteria for filtering out papers for the study

Exclusion Criteria	**Rationale**
Papers that are duplicates of papers already included.	Including duplications will skew the results of this review. If duplicate papers are found, only the latest version will be included

	and all others excluded.
Papers that are systematic literature reviews.	Systematic literature reviews that study other systematic literature reviews are considered tertiary studies. This Systematic literature review is a secondary study such that it reviews primary studies.
Papers that address Agile software development without Global or Distributed human resources	Unless a paper focuses on using distributed human resources for Agile Software Engineering, it was avoided.
Papers that address Global or Distributed human resources on non-Agile software development model	Unless a paper focuses on using distributed human resources for Agile Software Engineering, it was avoided.

4.4. **Study Quality Assessment**

Once the papers were analyzed using the inclusion and exclusion selection criteria, the remaining papers were manually validated to ensure the quality of selected studies. Quality assessment criteria listed in [68] were used to appraise the attributes of the research design and reporting of the selected studies.

The abstract was reviewed to understand if the problem participants, method of research, findings, and conclusion of the study were mentioned. Based on [68], the introduction sections of included papers were scanned and the problem definition, research questions, domain, and subjects were captured. Data collection and analysis were gathered to ensure research was based on quality data. Based on [68], the interpretation of analysis was reviewed to ensure all variables were accounted in the results. Since the focus of this study is to capture risks and solutions in detail, papers were assessed to ensure results were detailed, assumptions documented, and practicality of the study was focused on realistic team structures. Lessons learned were reviewed and gathered from the discussions sections as data could be used as part of DASE solutions. Threats and future works were reviewed as per [68][70].

A checklist was created and the reviewed papers were compared against the checklist to ensure quality (see Table 7).

Table 7: Quality Assessment Checklist

Area	**Criteria**
Abstract	Does the abstract describe: (1) the problem under investigation, (2) the participants, (3) the empirical method, (4) the findings, and (5) the conclusions.
Introduction	Is the problem defined? Are research questions documented? Is the domain of evaluation documented? Who has observed it (samples/instruments)?
Experiment Planning	Is data collection explained? Is data analysis explained?
Execution	Are interpretations of analysis explained?
Analysis	Were results explained in details? Were assumptions described? What are the practical implications of this study?
Discussion	Are interpretations of analysis explained? Where lessons learned mentioned?
Conclusion	Is there a concise summary of the research? Where threats described? Does the paper document future work?

The above questions were answered in Yes, No, or somewhat. A weight of 1, 0, 0.5 was assigned for each question for each paper that has gone through the inclusion and exclusion criteria.

4.5. **Data Extraction**

The data extraction form, shown in Appendix C, was designed to accrue all the necessary information required to address the research questions and quality assessment criteria. In addition to acquiring the information needed to address the research questions and quality assessment criteria, the following standard information was also extracted from each primary study: Title of the Paper, Sources (Database and Journal), Date Published, Paper URL, Document Object Identifier (DOI) and Authors.

The purpose of collecting the aforementioned information was to provide analysis of the meta-data of the studies themselves. For instance, distinguishing the time frames of the studies (i.e. how many studies were published in year 2007 versus year 2012). This measurement provided insight into the growth and interest in DASE research. Other points of interest that can be answered include who the main players are in DASE research, how readers can access the studies via URL or DOI, and what sources are more likely to publish DASE research, and more importantly, publish high quality research. However, this review has limited its work to reporting the findings associated with answering the research questions stated in Section 2.

As part of property 1, the introduction section of each paper was reviewed to get a better understanding of the context of the study. The problem being reviewed, ways it impacts an organization, its occurrence, subjects, and importance were reviewed and understood to answer RQ 1.1 and RQ 1.2.

Property 2 expanded on property 1 by understanding the reasons why organizations choose to engage in DASE, the lifecycle of the project where they utilize distributed human resources, and what type of distribution model was used. Human resource distribution model is an important variable, and as the results could vary if the team members are part of the same organization or a vendor organization. Additional factors that could affect results, such as human resource experience and possible collocation was also extracted. Data extracted was useful for RQ 1.1 and RQ 1.2

Property 3 focused on research method and study environment. Research methods include but are not limited to: case studies, surveys, and experiments [68]. Study environment is limited to: academic and industrial. Additionally, the goal of the paper and the subjects were captured. These provided us with a good idea of the research technique and how the different variables could have affected the study results. Data extracted were useful to answer RQ 4.1 and RQ 4.2.

Property 4 captured the overall documentation of risks, issues, and workarounds based on the Agile model used. Data extracted were useful to answer RQ 2.1 and RQ 2.2. Results showed how

the different Agile models impact issues faced during studies. In addition, reasons why a particular model was used were captured to get an understanding of justification.

Property 5 was used to extract data to answer RQ 3.1 and RQ 3.2. Based on the above variables, it was useful to see if the project was considered successful. There might be room for bias since it is more likely for researchers to publish successful projects. Should the project fail, it would be interesting to see whether an interest sparks among researchers and organizations to conduct further research or if organizations would engage in non-Agile projects with distributed human resources.

Below we provide further details on traceability between research questions and properties:

RQ 1.1 aims to bring forth the SDLC phase involved in distributed agile software engineering. This was answered by understanding the context of the study (property 1.1, 1.2), by analyzing reasons why this particular study engaged in DASE (property 2.1), the impacts of DASE engagement (property 1.3), and SDLC phase more active in DASE (property 2.2).

RQ1.2 reveals the human resource distribution model. The answer for this question required analysis of several points in each paper. Has the organization limited distributed human resources to a certain lifecycle (e.g., Development or Testing) (properties 2.2, 2.7), the type of distribution model utilized (property 2.3).

RQ1.3 attempts to understand organizations past experience in DASE. This was elicited by understanding human resources previous experience in Agile and working on distributed teams (properties 2.4, 2.5, 2.6).

RQ 2.1 aims at documenting risks and issues reported in projects that have engaged in DASE. This was accomplished by reviewing any risks and issues that were faced in the reviewed paper (property 4.1). Capturing risks and issues might have been the focus of the reviewed paper, so this information will be available throughout the document. Risks and issues could be dependent on the Agile model used (Scrum, XP, etc) and so, it was worth understanding why an organization engaged in that specific Agile model (properties 4.3 and 4.4).

The aim for **RQ 2.2** was to understand the workarounds or mitigation strategies that had been put in place. Details were captured by reviewing workarounds and mitigation plans documented in the papers (property 4.2). Workarounds or mitigation plans that were implemented to deal with the issues and risks along with those listed in the proposed solutions were documented.

RQ 3.1 aims to understand if an Agile model stands out as being the best when working with distributed human resources. This was accomplished by noting if a project was considered a success (property 5.1). Mapping this to property 4.3 gave an idea if one model leads to more success than the others.

The goal of **RQ 3.2** is to understand the failure rate between the different Agile models. This was based out of the information extracted as part of property 4.3 and property 5.1. With this data captured, it was interesting to further capture if there was interest in engaging in DASE again

(property 5.2). Additionally, for those that have failed, future interest of the organization to engage in distributed human resources on non-Agile projects was captured (property 5.3).

RQ 4.1 documented the main method used by researchers. This was captured based on information extracted as part of property 3.1. Additional data captured as part of property 3.2 and property 3.3 provided a holistic answer.

The goal of **RQ 4.2** is to get an understanding of the research environment. Research environment includes the subjects of the research, whether it was conducted in an academic setting or industrial setting, degree of realism, and the focus of the research. This was based on the data extracted as part of property 4.4. Degree of realism helped explain the maturity of the field as research within immature disciplines tends to be more exploratory in nature than research in mature fields that focus more on testing frameworks, practice, methods, or tools [72].

5. Conducting the Review

This section provides a description of how the review papers were selected for this review. Steps provided in the Review Methods Section were used to execute the search.

5.1. Inclusion and Exclusion of Studies

Initial query search and after snowballing resulted in 55 papers on IEEExplore, 186 on ScienceDirect, 118 on SpringerLink, 5 on Wiley Online, and 43 on ACM portal. Implementing the Inclusion and Exclusion criteria, as outlined in Tables 5 and 6, on papers resulted in a total of 75 papers remaining. The primary reason for exclusion was the coverage of both Agile methodology and Global / Distributed aspect. Of the remaining 75 papers, 12 papers did not meet quality standards. The quality of the 75 papers was assessed based on Table 7. The quality checklist required clear documentation of the problem, when it occurs, observation, and others as explained earlier. At the end, a total of 63 papers remained. The steps of the study inclusion process is shown in Figure 2.

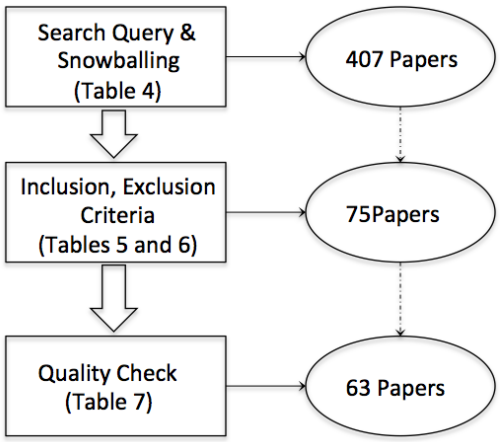

Figure 2 Inclusion Process and Results

The manual process for including or excluding studies and also performing the quality checks was performed as follows. The authors first collectively agreed on the search query to be used for identifying relevant publications from the aforementioned databases. The first author was then responsible for executing the query and retrieving the list of papers. The first author would then decide on the inclusion or exclusion of a paper from the study based on Tables 5 and 6. There was a collection of papers that could not be classified as include or exclude by the first author and were labeled as unclassified by the first author. The authors then reviewed the classification of the first author collectively. All the authors checked the clearly accepted or rejected papers into the study. The unclassified documents were then evaluated by the second author, which later confirmed his decision with the third author of the study. Once 75 papers were selected based on the inclusion/exclusion criteria, the authors evaluated the quality of the papers collectively based on Table 7. As a result 63 were accepted into the study at the end. The distribution of the final accepted studies in terms of year of publication, digital library and type of publication has been depicted in Figures 3 and 4. As seen, it seems that the major publications appear in conferences in this domain mainly published by Springer and IEEE.

5.2. **Threats to Validity**

The main threats to this study are the review protocol, paper selection, and data extraction. This section will further address each of these threats.

5.2.1. *Validation of the review protocol*

The review protocol developed for this systematic literature review was created prior to conducting the review. Several guidelines were consulted including the search protocols listed in [68], [69], and [70]. However, it was [68] that were the primary source of guidance.

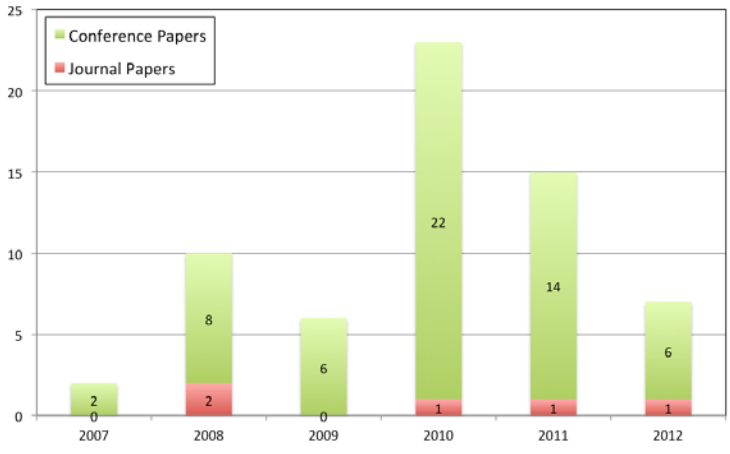

Figure 3 Distribution of Selected Studies by Type

5.2.2. *Validation of publication and primary study selection*

As mentioned by [68], bias in paper selection could be a result of publication bias. Publication bias refers to the problem that positive results are more likely to be published than negative results. To address this, there are several strategies that could be put in place including scanning grey literature, scanning conference proceedings, and contacting experts and researchers working in the area. We should point out that grey literature, such as organization white papers and lessons learned were reviewed manually to address bias in paper selection.

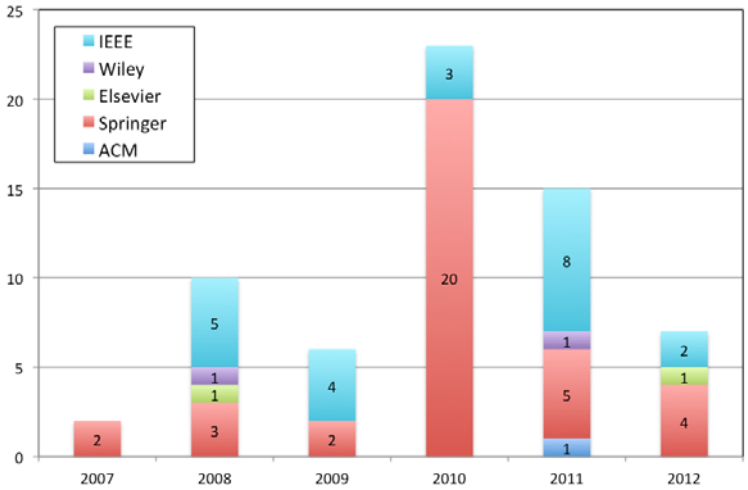

Figure 4 Distribution of Selected Studies by Digital Library

In order to prevent from selection bias, papers were searched for IEEExplore, ScienceDirect, SpringerLink, Wiley Online Library, and ACM. Both conference and journals were searched in the above. As such, selected papers were queried through a wide database. Multiple publications of the same data were also avoided, as duplicate reports would seriously bias results. In the event of duplications, the most recent study was used.

In order to validate the inclusion and exclusion criteria, random sets of five studies were reviewed based on the inclusion and exclusion criteria. The results were analyzed and validated by all of the authors. All 407 studies were subjected to the selection process. Using the criteria's mentioned in Section 4, 63 studies were deemed acceptable and tagged as selected. The remaining studies were either rejected or classified as related work. Reasons for acceptance and rejection were noted on all studies. The final results were analyzed and validated by all of the authors and considered to be acceptable.

5.2.3. *Validation of data extraction criteria and classification*

Data extraction criteria were described in detail in Section 4. The level of detail provided will prevent threats to validity of the results of this review. In some cases, published papers that are part of this paper's review could be written poorly, have ambiguous data, or do not include relevant data [81]. This makes data extraction difficult especially when fitting data into

enumerations. Hence, it was necessary to validate the data extraction properties against credible sources.

The data extraction property 1 was sourced directly from the primary studies reviewed in this study. Each study reported the context, problem on hand, and the scenarios in which it occurs. Therefore, at best, this information was reported verbatim based on information provided in the reviewed studies. In this circumstance, the extracted information was reviewed by all of the authors for verification. No disagreements in the extracted data or classification were reported back.

The data extraction properties 2 and 3, which help capture details of the project that had engaged in DASE, were based on the authors' practical experience and from early review of papers on this topic. Since the authors experience is not considered to be a credible source of information, reviewing published papers helped define the enumerations.

The data extraction property 4 and 5 was sourced by literature review performed by [71], [72], [73], [74], and [75]. Additionally, early review of papers helped define preliminary guidelines.

Data classification proved to be without certainty since the studies under review did not provide precise answers to the data extraction criteria. Many properties were not described correctly or mentioned at all. In these circumstances, [68] recommends contacting the author of a questionable study to assist in resolving uncertainties and provide clarity to unknowns. However, [82] provides an alternative suggestion to contacting authors, which allows for general impressions of subjective evidence to be made by the reviewer. In this paper, the option to make general impressions on subjective evidence was used. Again, in this circumstance, the extracted information was verified by all the authors. No disagreements in the extracted data or classification were reported back.

In order to avoid data extraction bias, it is recommended by [68] that two or more researchers should perform data extraction independently. Data from the researchers must be compared and disagreements resolved either by consensus among researchers or arbitration by an additional independent researcher. This was clearly taken into consideration and addressed as outlined in Section 5.1.

5.2.4. *Limitations*

The distributed agile software engineering community uses many different terminologies for the various techniques and approaches that are available and currently being used. Our attempt has been to devise a search query, as shown in Table 4, which is as inclusive as possible. However, it is possible that the use of additional keywords such as *lean, outsourcing, offshoring,* and the like could have expanded the search space. We note this limitation and would like to point out that the primary studies selected in this systematic literature report is based primarily on the used search terms, namely (Global or Distributed) AND (Agile OR Scrum OR XP OR Pair Programming).

6. Results and Analysis

This chapter provides a discussion and analysis surrounding the results of this systematic literature review based on the 63 primary studies selected. The discussion is structured based on the research questions presented in the Background section.

6.1. What are the conditions under which organizations choose to adopt DASE?

This question aims to answer details of the current research context and to get an understanding of the circumstances surrounding engagement in an Agile development model using distributed human resources.

6.1.1. *What phase of the SLDC has utilized distributed human resources in Agile (RQ 1.1)?*

This question helps us understand which SDLC phase primarily uses distributed human resources. Review of the published literature indicates that projects in 84% of the papers had completely integrated agile in DASE using distributed human resources. In Table 8, "All Phases" refers to those papers that reported they had completely undertaken a DSD project using Agile principles. However, there were also other papers that had mentioned only selected SDLC phases as their target phase were they applied their approach. Our major finding is that projects that engage in DASE decide to roll it out throughout all of the SDLC phases. This is an expected finding given the iterative and rapid nature of Agile practices, as it would be rather difficult if not impossible to deploy an agile strategy in one of the SDLC phases in isolation. The papers not classified under "All Phases" in Table 8 are those that explicitly mention that they have only contributed to one of the listed SDLC phases in Table 8; therefore, Table 8 lists the SDLC phases as their were mentioned in those papers. Our investigation showed that such papers are mostly focused on modifying specific phases of the traditional software development lifecycle using the concepts of iterative and incremental progression, and distributed software development.

Table 8: Software Development Lifecycle

Phase	Papers	Percentage
All Phases	53	84%
Planning	4	6%
Development	2	3%
QA and Testing	2	3%
Requirement	1	2%
Design	1	2%

6.1.2. *What is the typical human resource distribution model (RQ1.2)?*

This question provides an idea of the number of teams that were part of a project that undertook DASE. Few of the 63 papers reviewed had performed research on multiple projects as part of their publication. This is the reason why the number of projects in the tables can be more than 63. Peak time zone between teams was also analyzed to get an understanding of how far the teams were.

Table 9 summarizes the number of teams used in DASE projects. More than a half of the projects (53%) reported in the included papers had engaged in DASE using two teams, while 11% had three teams. The primary reasons to engage with distributed human resources was to save cost and access talent [36][37]. This is primarily done by engaging an additional team [37]. It was noted in one case where 12 teams were engaged on a project [42]. The scope of that project involved creating complex software for a multinational software development firm.

Table 9 Number of Teams

Sub Teams	Total Projects	%
2	36	53%
Not Mentioned (or N/A)	15	22%
3	8	11%
4	5	8%
5	1	2%
6	1	2%
12	1	2%

Table 10 summarizes the time difference between teams. In terms of time difference between teams, two projects [18][30] were implemented using distributed human resources – both in the same time-zone. These projects were Academic and were implemented using Students. Three projects [14][44][52] had a time difference of 1 to 3 hours. These projects were implemented within Europe. 25% of the projects had teams that were (at peak) 3 hours to 5 hours apart. About 37% of the projects had not provided their team breakdown, and as such, we were not able to use results for analysis.

Table 10 Time-zone difference

Time Difference	Total Projects	%
Not mentioned	25	37%
3 hours to 4:59 hours	17	25%
9 hours to 10:59 hours	9	14%
5 hours to 6:59 hours	6	9%
7 hrs to 8:59 hours	5	8%
1 hour to 2:59 hours	3	4%
0 hours	2	3%

The distribution of the number of teams and the peak time zone between the teams are two important factors that can show how deeply DASE has been deployed in practice. As shown in Tables 9 and 10, from amongst the projects that reported these data, the majority of the projects were inclined towards a smaller number of teams e.g. two or three teams and also the peak time zone difference was mainly restricted to teams that would have at least some minimum work hour overlap. One of the main reasons for this could be to alleviate issues of communication and

coordination, as we will show in Section 5.2. It should be noted that the distribution of agile practices and distribution types has already been reported by Jalali and Wohlin (see Figure 4 of [73]).

Table 11 provides us with an idea of why organizations choose to engage in DASE. 52% of the projects had engaged in DASE as it is part of their business practice. These organizations or their vendors had already engaged in Agile using Distributed human resources in the past. 12% had implemented DASE for experimentation. These were primarily Academic projects. 5% had engaged in Agile using Distributed human resources because they had heard of the benefits of Agile or distributed human resources. 6% had engaged in Agile using Distributed human resources to simulate real world scenarios. These projects were either Pilot projects or projects to assess the feasibility prior to full engagement [4][5][8][25]. Organizations chose to perform feasibility first to ensure that they have the capability to handle Agile using distributed human resources. In the case of [4] and [8], organizations learned of how to streamline their processes on future DASE projects.

Table 11 Reasons for engaging in DASE

Time Difference	Total Projects	%
Business Practice	35	52%
Not mentioned (and N/A)	17	25%
Experiment	8	12%
To simulate real word experience	4	6%
Heard of Agile	3	5%

6.1.3. How much experience do human resources have in Distributed Agile Software Development (RQ 1.3)?

This question aims at understanding the knowledge that human resources from the engaging organization have with regards to distributed and Agile development. Table 12 summarizes experience of human resources engaged in DASE. As part of the review, it was noted in 38% of the projects that most team members had experience in DASE. In 14% of the cases, human resources did not have experience. In 11% of the projects, some human resources working on the projects had experience in DASE.

Table 12 Experience in DASE

Experience	Number of papers	%
Yes	24	38%
Not mentioned (and N/A)	23	37%
No	9	14%
Some Resources	7	11%

Table 13 summarizes the experience of human resources in distributed models. Digging deeper into human resource experience, it was noted that in 44% of the projects, majority of the human resources had experience working with distributed human resources. Such experience could have been in Agile or non-Agile projects. In 10% of the projects, human resources did not have experience working with distributed human resources while in 13% of the projects some of the human resources had experience working in a distributed model.

Table 13 Experience in distributed team structure

Experience	Number of papers	%
Yes	28	44%
Not mentioned (and N/A)	21	33%
Some Resources	8	13%
No	6	10%

When it comes to Agile development practices, in 40% of the projects, team members had previous experience with Agile while in 15% of the projects, human resources had not worked with Agile methods in the past. In 10% of the projects, some human resources had experience working with Agile methodologies. Table 14 summarizes the experience of human resources with Agile methods.

Table 14 Experiences in Agile

Experience	Number of papers	%
Yes	25	40%
Not mentioned (and N/A)	22	35%
No	10	15%
Some Resources	6	10%

Working in Agile practices requires human resources to work face-to-face, but collocating is difficult when working with distributed human resources. It is possible to facilitate collocation through different strategies. One is to allow for the distributed team members to get together at the start, which is known as seed visits. The other strategy would be to allow team members to have face-to-face meetings at different time intervals of the project, which is known as maintaining visits or a combination of seed and maintaining visits. It was noted that in 16% of the projects, human resources met in the beginning (seed) and continued visiting throughout the course of the project (maintaining). In 44% of the projects, human resources did not collocate. In 10% of projects, human resources collocated in the early phases / iterations / sprints (seed visits), and in 13% cases, human resources met during the course of the project through maintaining visits. Table 15 summarizes the collocation strategies implemented on DASE projects.

Table 15 Collocation during project lifecycle

Did team member collocate?	Number of papers	%
No	28	44%
Not mentioned (and N/A)	11	17%
Yes (seed and maintaining)	10	16%
Maintaining Visits	8	13%
Seed Visits	6	10%

6.2. **What are the biggest threats when adopting DASE (RQ 2)?**

The answer to this question provides details on risks documented in DASE and solutions listed to deal with risks. Risks were categorized under 1) communication, 2) collaboration, 3) coordination,

and 4) cultural differences. This categorization of risks in DASE under these four classes is based on work done by [7][10][13][15][27][34][39][45].

6.2.1. What are some of the biggest risks in DASE (RQ 2.1)?

This question aims at documenting risks that projects have faced and issues that have risen during the course of the project.

Communication

In communication, time zone differences, lack of synchronous communication, language differences, infrastructure (e.g., video conferencing or tools), and lack of visibility on priority, requirements, demo, and sprint reviews were considered the big risks in DASE.

Time zone differences caused major challenges when it came to arranging meetings, especially longer meetings as it related to Sprint planning [1][55]. In cases where time difference was more than 5 hours, human resources had to arrange a common time for meetings (usually late or early hours). Additionally, due to time zone difference, communication was mostly indirect via email and via comments in a software system, if available [3][8][60]. Telephone was considered a good choice, but was found effective only if both parties knew each other already. Since team members do not have a chance to get to know each other personally, people hesitated to initiate direct contact and preferred to communicate indirectly [24]. This led to a *single-point-of-communication* way of information exchange: the project leaders [24].

Teleconferencing was utilized in some meetings, but was listed as being exhaustive due to sound quality, difficulty in recognizing speaker's voice, and language differences [1][37][53][61][62]. In some cases, the network connections between offices were not fast enough for videoconferencing or offices did not have video conferencing capabilities [46].

Although in most cases, team members spoke a common language (albeit not the first language for the involved parties), it was noted that speaking style contributed to *language issues*. As an example, some team members can be loud and direct while other team members can be careful and cautious in their expression [27] [33] [39].

Since most communications were conducted by phone and (in some cases) video conferencing, considerable time was spent in projects clarifying items being discussed, because much of the meaning, tone and emotion were lost through this communication medium [2][37][39]. As such, participants did not get a clear understanding of the requirements and priority. In some cases, the facilitator *lacked enough experienced* to understand if the team had understood the requirements correctly and that the right priority is assigned to this activity [4][7][11][17][34][57][59][62].

When it comes to reviewing Demo and Sprints with all teams involved, software were not always used to demonstrate Sprint reviews. This is because not all organizations had an *infrastructure set up*. A technically savvy Product Owner made an effort to take screenshots and videos of the product to share with the team, but distributed team members found it hard to follow [8][53].

Collaboration

In collaboration, availability of the Scrum Master, human resource knowledge, inexperience, transparency, human resource coaching, mentoring, motivation, trust, productivity, lack of team structure, improper work distribution, and lack of strategic solutions were listed as some challenges.

One of the concerns under collaboration was that Scrum Masters did not facilitate Scrum Reviews [8]. This led to delays and absence of working software to demonstrate. Additionally, with the lack of video conferencing, team members did not have visibility as to who is the Scrum Master in the tool. With inexperience in Agile methods, projects missed a strong leadership from Scrum Masters during negotiations with business partners who flooded team members with a long list of last minute changes by the end of each sprint. *Project management maturity* on agile practices impacted development life cycle [8].

It was noted that on some projects team members *lacked knowledge of Agile methods*. This was most noted when it came to Scrum Masters not having the knowledge to drive their team [9][10][16][28][34]. Additionally, clients that did not have previous experience with Agile methods relied on the project manager who tried to work with the client as a Meta Scrum Master / coach to bring the organization into an Agile way of working and acted as proxy product owner. This resulted in more issues in cases where the representative was not experienced enough with domain knowledge to interpret customers' needs and devise better solutions [34]. Teams that worked with a remote Scrum Master and/or Product Owner were impacted on days when there was misunderstanding in scope [51].

Another area that becomes challenging in DASE is coaching. When projects are close to a catastrophe, the coaches come into the picture. When *coaches are remote*, coaching is not very effective [19][45]. The biggest problems in the project involved multiple sites are lacking and poor functioning processes and the lack of collaboration between the sites. Thus, solving these problems by coaching only one site is impossible. To cover the gap of a missing local mentor, other managers took over the local mentor role, which did not help [52].

Lack of Team structure and Roles and Responsibility is another challenge in DASE [9][45][55]. This happens, more frequently, when team members lack experience. Agile practices state that every team member must collaborate as a generalist in project tasks [9]. This only works if there is information flow between teams. It was noted that customer organizations were reluctant to openly share information with the contractor or vendor organizations, even though they were implementing the same system [45][53]. The detached nature of the customer and its representatives manifested itself especially when requirements that the remote team was accountable for were discussed [48].

Trust and lack of productivity is another challenge. During project implementation, trust needs to be established and maintained; otherwise, remote team members will not be able to get along with

each other [35][47][59]. It is difficult to foster team bonding and collaboration with the distributed teams with few or no face-to-face interaction.

Work distribution with distributed human resources is another challenge. Distribution of work is not easily done with distributed human resources. This is because dependencies required collaboration between team members that are working toward implementing the same stories [10][16][23][31][40][56]. Additionally, some work cannot be distributed due to remote access challenges. At such times, the Scrum Master, Project Manager, and Leads should be capable of foreseeing such dependencies and dividing work when possible. Sometimes, due to time pressure, user stories from a single feature are often distributed and are implemented by multiple teams [15][32][37]. This in turns increases the amount of collaboration. If processes are set up, then unique responsibilities should be assigned.

Another challenge noted was the team *missing the big picture* and making tactical decisions. Agile teams do focus more on tactical rather than strategic decisions [42][62]. The reason why is that working in time-boxed iterations gave teams a short-term focus, usually of 2–4 weeks in duration. As such, teams lost sight of the organization's goals for customer delivery and how their decisions helped reach those goals.

Coordination

Under coordination, lack of documentation, cost for synchronous communication, shared components, sharing of proprietary or sensitive data, and lack of process were considered as major challenges.

Agile, unlike Waterfall, does not focus of full documentation of requirements or product and sprint backlog [37]. As such, teams were, at times *unclear on the requirements* that needed to be implemented. Requirements were gathered on exhaustive meetings (8 hours duration) and documented in minutes [9][18]. Index cards were posted on walls at the office were requirement reviews were held and not always replicated manually at the other sites [25][51]. Story cards from one site are not directly shown to the distributed teams, and key behaviors, such as modifying index cards are difficult to share with remote colleagues.

Cost of synchronous communication is another challenge under coordination. Due to time zone differences, teams had to arrange a common time for meetings [29]. This common time usually was early or late during the date, resulting in the team member working outside of regular working hours. This resulted in cost increase as there was a change in working hours.

Having common or shared components was listed as another challenge as it makes coordination difficult. The Solution Architect designs systems based on organizational architectural direction and industry standards. Such design could, at times, have dependencies between components [7][42][53]. This leads to increased dependencies among products and components – with componentization being weak and code reuse being highly valued [42].

When dealing with very sensitive customer data, it was difficult for teams to pass data for testing or defect analysis [20]. While customers were willing to share those data with the primary team for the limited use in testing the software under development, the agreements generally did not extend to offshore partners [20][40]. This limited some of what the vendor organizations could develop and test.

While processes were enforced on the primary organization, in dispersed teams it is even more important to have supportive processes defined than in organizations that work in one location. It was deemed crucial for project managers to clearly define artifacts to be developed and to assign unique responsibilities [8][24][30][36]. This caused confusion within teams as vendor organizations were not used to the processes enforced by customers.

Cultural Differences

Work practices, regional holidays, ways of speaking, hierarchical, and importance to project timelines were challenges under cultural differences.

Work pattern of human resources vary depending on where collaborating parties reside. For example, Indian developers remained mostly silent at the daily meetings and are instead engaged in forced communication by the Scrum Master [39]. Additionally, it was noted that human resources in some countries required work to be assigned to them as opposed to human resources suggesting what work they would like to be involved in [61]. This led to uncritical or sometimes boring tasks assigned to offshore teams.

Regional holidays result in a team of human resources being unavailable for project work. If the Project Manager is unaware of such holidays while planning, this could result in deviations from the schedule [5].

Language and practice are another area where cultural challenges apply. While communicating, it is noted that some cultures speak load and direct while some cultures are careful and cautions [7][11]. Additionally, some speak fast while some speak slowly. In some cultures, it is not acceptable to say No to family superiors – even if what is being said is wrong. All of these lead to gaps that are difficult to manage if team members are not aware of [47].

Another cultural problem, especially with the daily Scrum meeting, was the notion that human resources were "Reporting to Scrum Master" instead of synchronizing knowledge between colleagues [13]. This creates a false sense of hierarchy that some human resources are used to having – similar to having work assigned.

In regards to project timeline and milestones, some cultures consider the project schedule as guidelines as opposed to commitment. This results in misunderstandings and unset expectations set to the customer [42].

Table 16 summarizes risks and challenges documented in published literature. Risks and challenges are sorted by times reported.

Table 16 Risks and Challenges

Risk/Challenge	Communication	Collaboration	Coordination	Cultural differences	Times Reported	Percentage
Time zone	✓				26	9%
Knowledge / Inexperience / Misunderstanding / Transparency		✓			25	9%
Priority, Requirements, etc	✓				25	9%
Synchronous	✓				22	8%
Team structure / Roles and Responsibility		✓			17	6%
Lack of documentation (requirement, QA Plan, backlog, etc)			✓		17	6%
Work distribution		✓			17	6%
Trust / Productivity		✓			15	5%
Work practices				✓	15	5%
Infrastructure	✓				14	6%
Lack of processes			✓		11	4%
Language	✓				10	4%
Demo / Sprint reviews	✓				10	4%
Cost (for synchronous communication)			✓		8	3%
Speaking (loud/soft/do not say No)				✓	8	3%
Hierarchical				✓	7	3%
Project timelines/milestone				✓	7	3%
Scrum Master availability		✓			7	3%
Shared components			✓		7	3%
Coaching / Motivation		✓			4	1%
Proprietary / Sensitive data / Remote access			✓		3	1%
Holidays				✓	3	1%
Big Picture (time boxed sprints result in tactical decisions)		✓			2	1%

Table 17 Workarounds and mitigation plans

Risk/mitigation	Communication	Collaboration	Coordination	Cultural differences	Times Reported	Percentage
Infrastructure (Telephone, Video conferencing, Webcam, Email, Chat, etc)	✓				33	12%
Face-to-face visits	✓				28	10%
Encourage formal and informal communication	✓				27	10%
Tools		✓			21	8%
Training inexperience resources		✓			20	7%
Communication strategy	✓				17	6%
Ambassadors (coaches, governance)	✓				14	5%
Overlap work timings		✓			14	5%
Documentation (formal documentation)		✓			14	5%
Review lessons learned / feedback from previous sprints/projects		✓			12	4%
Monitor work progress		✓			9	3%
Short iterations/sprints		✓			8	3%
Architecture Centric / Modularized (involvement to break/decrease dependencies)		✓			8	3%
Distributed Scrum-of-scrum (ambassadors)		✓			7	3%
Interview resources before hiring				✓	7	3%
Follow-up questions on requirements				✓	6	2%
Keep some work local (e.g. Proprietary)		✓			6	2%
Daily builds		✓			6	2%
Decentralize decision making		✓			6	2%
Smaller teams (per story)		✓			5	2%
Planning around regional holidays		✓			1	< 1%

6.2.2. How are risks, limitations, and mitigation strategies in DASE dealt with (RQ 2.2)?

The aim of this question is to document workarounds or mitigation strategies that projects have utilized to deal with the risks and issues, as they were uncovered.

Communication

In order to work around communication related issues, some solutions mentioned were having a good communication infrastructure, encouraging teams to engage in both formal and informal communication, creating and enforcing a communication strategy, having ambassadors, coaches, and centralized governance, and having face-to-face visits.

Having teams *use telephones, video conferencing, and webcam* during personal meetings, emails, and internal chat are some ways to enhance communication [1][2][17][44][52][56][57]. During Scrum meetings, teams could use video conferencing and utilize screen sharing when possible [1][61][63]. There could be multiple Scrum meetings – one internal to sites and the second with all

teams. Having a good infrastructure was listed as a work around in several papers and is the most important way to improve communication between teams. Teams looking for a free solution had utilized Skype video call for Scrum meetings and LiveMeeting or WebEx for Demo presentations [53]. Separate meeting rooms were set up at each site with stories posted up on Walls. As part of collaboration, team members were encouraged to use email to communicate when work hours did not overlap [45][53]. When, and if, there was an overlap in work hours, team members were encouraged to use telephone and webcam or internal chat software. Meetings could be conducted in various ways. The first solution is to use video conferencing technology, the second was each location in a conference room, and the third was conference calls with headsets from individual desks [46]. The solution that worked best, but still had issues was using headsets for all team members and doing a conference call from each member's desk, as a kind of virtual conference room [17]. In such a setting, team members had good quality due to the use of headsets, webcam to have a personal touch (although not easily used with a large team), and access to desktop sharing to ensure all human resources were looking at the same thing. Same etiquette of only one person at a time talking was followed. With everyone on the phone using a headset, each person had the same experience and quickly learned to allow another person to finish a statement before speaking themselves.

Encouraging formal and informal communication is another way to improve communication between teams. Using tools such as an internal wiki and emails should be used when asynchronous communication channel makes sense [29][51][53]. Wiki was found to be one of the most useful communication channels in the implementation phase of the project largely due to the distributed nature of the effort [45][51][57]. Wiki-based agile planning tools can also be utilized to publish, manage, integrate and distribute agile planning information [51]. The advantage of using Wiki-based systems is that they provide a plain environment, making it easy to check project status, update task lists and view the team members' work progress. Wikis are an asynchronous platform for agile developers' communication and, thus, mostly helpful for progress tracking. Informal meeting helps to relax minds and build better relationship between human resources that collaborate [48].

Creating a *Communication Strategy* for a project helps define a set of scheduled or event-driven communication activities along with a mapping between these activities and communication media to be used during their execution [2][3][18][53][59][61]. An example of communication activities is an XP project that is undergoing the planning game or daily stand-up meetings [49]. The goal of a planning game is to get prioritized requirements from the customer [18][28][49]. The goal of a stand-up meeting is to get everybody in the team up to date on the current status including information about problems and solutions. Each communication activity has different needs for communication media (e.g.,: LiveMeeting, Video Conferencing, and Wiki) that facilitates it.

Ambassadors (or coaches or a governance body) are dedicated human resources with the task to bridge between remote teams. It was suggested that Ambassadors should concentrate especially on facilitation of communication between the sites by helping in solving problems and finding the

right persons to answer questions coming from the customer [5][11][13][17][18][29][34][52][58]. Additionally, such resource can also help resolve misunderstanding and help in language difficulties by communicating in the language that the customer understands.

One of the best ways to *improve trust* and help collaboration would be to *collocate*. Since this is not feasible when working with distributed resources, it was recommended to have teams meet at times during the project [11][19][31][44][53]. There were two types of face-to-face meetings used – Seed visits and Maintenance visits. Seed visits were visits where teams (or some team members) collocated for the initial iteration/sprints of the project. Maintenance visits were visits where teams (or some team members) collocated for brief periods throughout the course of the project. Both types of collocation strategies helped increase team comfort and helped establish trust. In some cases, organizations tried to utilize conferences where team members could meet.

Cultural Differences

There are ways in which cultural differences could be dealt with. By following up on questions to ensure team members have understood and by interviewing resources prior to engaging them on projects are two ways in which risks could be mitigated.

To avoid miscommunications or misunderstanding of requirements in the DASE process, *numerous feedback loops* were put in place in numerous projects [2][28]. In essence, remote team members used the daily Scrum meeting to update the team on what was completed on the previous day and what was planned to be completed. This raised the customer's confidence that the team have understood the scope correctly [11][25][33][34][48]. Additionally, the customer (or a representative) or the Business Analyst would ask follow up questions to ensure that the team has a solid grasp on the requirements.

It was noted in some projects that resources were unfit to perform assigned activities. In some countries available talent base is large but the true skill set of a given individual often varies from the picture presented by their resume [36]. Papers suggest that a rigorous recruitment process should pay attention to both technical competence and cultural fit [36].

Collaboration

There are a number of proposed solutions in place to deal with Collaboration related risks – overlap work timings, monitor work progress, review lessons learned, planning around regional holidays, training resources, keeping some work local, utilizing tools, daily builds, shorter sprints, decentralizing decision making, documentation, smaller teams (or teams per story), creating a modularized architecture, and using a Scrum-of-scrum model.

Overlapping work timings is an easy way to have formal and informal communication. In some cases, such as those with greater than 7 hour time zone difference, overlapping work timings will be very expensive and not feasible [3][7][19][27][33][39][53]. Having the team overlap timings by having resources come in very early or stay late will be bad for morale and expensive for the project (if overtime is paid). In cases where time zone is less than 7 hours, it is possible to have

some team member start work early on one site and late on the other. This allows for team members engage in synchronously communication [19][35][51][53].

Monitoring work is another way of dealing with issues. Iteration/sprint review sessions and daily scrum can be used to monitor work progress [3][24]. Additionally, senior resources can assist with code reviews, test case execution, etc. to ensure that resources are performing per expectations.

Reviewing lessons learned from past DASE projects and previous sprints can also be useful to improve on mistakes made in the past. It was recommended to document lessons learned after each sprint/iteration to ensure that future iterations can improve on inefficiencies [17][32]. This will lead to overall improvement and an improvement to the quality of work in future sprints.

Although planning around regional holidays is not a major item, it is important to note that if regional holidays were not incorporated in the project schedule, there would be delays to the sprint [5][46]. Additionally, if a regional holiday falls on a demo or sprint planning day, then the Scrum Master and/or Project Manager will have to coordinate with other teams to find a suitable day to perform those activities.

Resources could be inexperienced in their activities or in the DASE process. In either case, resources need to undergo training, mentoring, or coaching. It was highly recommended that if the Scrum Master was inexperienced, then the project team should *replace the Scrum Master* with a more experienced resource [5][11][22][34][36][44][57]. Projects that miss a strong leadership from the Scrum Master end up being flooded with last minute change requests from the business partners. It was also recommended to have employees undergo a *one-week training course* that explains Agile, distributed team structure, and processes [11][36][44].

While most development work can be distributed, there is project work that is not easily done in a distributed way. It is recommended to keep such work within a team. Some examples of suitable candidates to keep local would be proprietary work, work that cannot be done due to remote access restrictions, testing using data that cannot be shared with unauthorized team members, or work that is considered complex [20][42][58].

Tools, if used correctly, can ease project coordination. This is especially true in DASE [7][11][46]. Tools can be used to document and easily perform activities such as share requirements, design, development, test cases, data, and infrastructure details. In addition, there are tools such as Wiki, Whiteboard, Sharepoint, and ScrumWorks (for backlogs) that can be used to provide team members a digital forum using which they can fill the gaps introduced by having distributed resources [7][11][46].

Daily builds are another way of improving the DASE experience. Releasing as many builds as possible, a project team can eliminate wastes in terms of waiting for a whole package to be tested [58].

Every sprint delivered not only increased team motivation but also improved collaboration and engagement. As such, it was recommended to have *shorter sprints* where something tangible is

made available to the customer [23][56]. It was also noted that overall quality, interaction, cooperation, and experience improves with every sprints. Applying short iterations, frequent builds and continuous integration in the development process leads to feedback. This feedback motivates developers and motivates developers to feel more like a team. A shorter development cycle can also be used to reduce risks and increase feedback for other teams [20][22][23][44][46][53]. If iteration time is shorter, more communication is needed to make sure next priorities in a product are being prioritized correctly.

Decentralizing decision-making is another way of improving the DASE experience. Because of this aspect, both geographical and temporal distances become less of an issue since developers can take certain decisions without having to confer with management, which could be located in another part of the world [12][20][41][49]. Additionally, the idea of applying semi-self-organized teams serves as a motivational goal [20][41].

Working with Agile methods with distributed resources requires formal documentation. Business should focus on artifact creation especially documentation from the product team. It was noted that by creating documentation, teams did not communicate as much, which worked to the benefit of the project since communication is considered a challenge when working with distributed resources [9][24][27][37][49]. Any documentation that would decrease communication issues in an Agile method should be part of the process.

Having teams set up such that one (collocated) team works on a story together was considered best practice [31]. Doing so decreases communication and dependencies between distributed teams.

Architecture-centric software engineering focuses on minimizing the inefficiencies associated with traditional process-centric development. The approach adopts a set of principles that are different and often initially uncomfortable in corporate contexts [21][32]. The key enabler for architecture-centric software engineering is to minimize dependencies between components. Although this is central to architecture design, architects often de-prioritize decoupling to achieve other attributes. Architecture-centric software engineering removes so many inefficiencies from the software development process that the output of the organization is much higher [32].

Table 17 summarizes workaround and mitigation plans documented in published literature. Workaround and mitigation plans are sorted by times reported.

Now given the risks and mitigation plans have been covered across all of the selected studies in Tables 16 and 17, it is interesting to point out what were the most challenging risks and the associated mitigation plans that were recommended based on the degree of experience the team members had. Team members' experience with DASE has already been reported in Table 12. We look at the three classes of experience defined in this table (Yes: experienced with DASE, No: not experienced with DASE, and Some Resources: some team members had experience) and report the top three risks and mitigation plans that were reported in each class.

In those studies where the team members were deemed to be experienced, the top three risks were as follows: 1) Communication – Infrastructure, 2) Collaboration – Inexperience, and 3)

Communication – Synchronous. It seems that in such teams the most challenging risks pertain to logistics of procuring the right infrastructure for communicating in a distributed environment (communication infrastructure and synchronousness). Collaboration – Inexperience referred to the effort required to train inexperienced human resources. In terms of mitigation strategies 1) Communication – Infrastructure, 2) Collaboration – Tools, and 3) Communication – Communication Strategy were deemed most important, which again reflects the need to effectively address the communication logistics within a distributed environment.

For the studies where the team members had no experience with DASE, the three top risks included: 1) Communication – Time Zone, 2) Collaboration – Team Structure / Roles and Responsibilities, and 3) Communication – Synchronous. It seems that teams with no experience in DASE struggle with synchronization issues such as time zone differences and the need to work under different conditions in a distributed asynchronous environment. In order to mitigate the issues, these papers reported 1) Communication – Face-to-Face Visits, 2) Collaboration – Overlap Work Timings, and 3) Communication – Encourage Formal and Informal Communication as the mitigation strategies. These mitigation strategies mostly try to address the need to effectively communicate under asynchronous conditions.

Finally, in the third class where some of the human resources had experience in DASE, the types of reported risks were similar to the risks reported in the class with no experience: 1) Communication – Time Zone, 2) Communication – Priority, and 3) Communication – Synchronous. The mitigation strategies consisted of 1) Communication – Face-to-Face Visits, 2) Communication – Encourage Formal and Informal Communication, and 3) Collaboration – Documentation. Analogous to when team members did not have experience the focus here is on issues of synchronization.

In brief, based on the literature, it seems that teams with more experience in DASE have concerns regarding logistics of communication and try to overcome this through infrastructure support, while less experienced teams face issues of effective collaboration and synchronization and therefore, employ mitigation strategies such as face-to-face meetings to overcome them.

6.3. What model of the Agile methodology is most adopted in DASE (RQ 3)?

The answer to this question will provide an understanding on the final outcome of the project and if certain Agile models stand out as being more successful.

6.3.1. Has one Agile model resulted in more success in distributed teams (RQ 3.1)?

The goal of this question is to understand the success rate between the different Agile models. Table 18 summarizes Agile models used in projects. Based on the 63 papers reviewed, 40% of the

project had used SCRUM. 14% had tailored and created a custom methodology called SCRUM and XP. 14% projects had used XP. One thing to note is that ScRumUP, a custom methodology, was created by one organization [6].

Table 18 Agile model used

Agile Model used	*Number of Papers*	*%*
Scrum	25	40%
Agile, specifics not mentioned	16	25%
Scrum and XP	9	14%
XP	9	14%
Pair Programming	2	3%
ScRumUP (custom)	1	2%
Lean development and Scrum	1	2%

In term of success, it is difficult to provide meaningful analysis as none of the projects was reported to have failed. Table 19 summarizes Agile models against success reported. Three projects (3% of total) were somewhat successful in a sense that the projects were complete, but with variation to scope, time, or budget.

An interesting observation that we would like to report on is the relationship between the risks and mitigation strategies that were reported in Tables 16 and 17 and the agile method that was reported in the papers. In other words, we were interested to see whether the agile method that was used as a part of each paper had any relationship with certain types of risks or not. Our finding was that besides pair programming, ScRumUP and lean development, which have less than 2 papers each and therefore the evidence is not conclusive, the other 4 methods mentioned in the literature typically had the same frequency in reporting the set of risks and mitigation strategies as shown in Tables 16 and 17. In other words, we could not see a trend where a majority of the papers related to a specific agile method were related to certain risks. One possible explanation for this could be that the risks and the mitigation strategies that were reported were mainly focused on the distributed aspect of DASE as opposed to the agile method that was used.

Table 19 Success and failure rate

Agile Model used	*Success*	*Failure*	*Somewhat*	*Not Mentioned*
Scrum	14	0	1	10
Agile, specifics not mentioned	6	0	0	10
Scrum and XP	6	0	0	3
XP	5	0	1	3
Pair Programming	2	0	0	0
ScRumUP (custom)	1	0	0	0
Lean development and Scrum	1	0	0	0

6.3.2. Is one Agile model shown to be worst in distributed teams (RQ 3.2)?

The goal of this question is to understand the failure rate between the different Agile models. Based on Table 19, none of the papers reported their approach had resulted in failure. In other words, all papers either explicitly mentioned or implied that their model for adopting DASE resulted in successful outcomes. One reason for such results is that maybe the community is inclined towards the publication of only successful project reports; therefore, papers included in this study only contained successful report and no failure reports were observed. It should however be noted that that 26 out of 63 papers (41%) did not explicitly indicate if their project was a success or failure.

It was noted that 44% of Scrum projects, 63% of general Agile, and 44% of XP projects did not explicitly report success. Of the remaining tailored methodologies, 33% of Scrum and XP hybrid did not explicitly report success while ScRumUp and Lean and Scrum tailed models had not reported failure. Based on this observation, it is difficult to make a definitive conclusion regarding the more effective and less effective models. However, based on the available data, it can inconclusively be said that tailored methodologies have a higher rate of success. It should be noted given the fact that these methodologies are specifically tailored and reported that they may suffer from reporting biases as well. We highlight in the future works section of this paper that we recommend that better and more substantial reporting rigor be used in the future to report on success and failure of the methodologies when used in practice. Furthermore, the reporting of failures should also be encouraged to allow the identification of the roots and causes of failure in DASE.

6.4. What is the strength of evidence in supporting the findings of the above questions (RQ 4)?

The goal of this question is to provide strength of evidence in the answers provided to RQ 1, RQ 2, and RQ 3.

6.4.1. What is the source of evidence (RQ 4.1)?

The goal of this question is to document the main data collection method used by researchers. The results in Table 20 shows the most common data collection method used in Distributed Agile Software Engineering research are observations represented by 33 papers (52%) followed by interviews in 19 papers (30%). In 3 papers (6%), a combination of observation and interviews were used while in 3 papers (5%), a combination of research and reviewing documentation were used. When it came to reviewing documentation, researchers reviewed sources such emails, communication logs, and Wiki to capture data. Experience and surveys were used in 1 paper each, at 2%

Table 20 The Data Collection Method Used

Research Model	*Number of Studies*	*Percentage*
Observation	33	52%
Interviews	19	30%
Observation and Interviews	4	6%
Research and Documentation	3	5%
Not Mentioned (and N/A)	2	3%
Experience	1	2%
Survey	1	2%

6.4.2. What is the data collection approach followed (RQ 4.2)?

The goal of this question is to get a better understanding of the research subject, area, degree of realism, and focus. The focus of current literature and degree of realism will help define the maturity level of the field since research within immature disciplines tends to be more exploratory in nature than research in mature fields that focus more on testing hypothesis, methods or tools [72].

A significant number of works were conducted in the industry represented by 51 papers (81%) while 8 papers (13%) were conducted in an academic setting. Equally, the number of studies used employees as subjects represented by 51 papers (81%), while 8 papers (13%) used students. The studies that used students as subjects typically recruited volunteer graduate level students to participate in a joint assignment.

It was noted that 65% of the papers captured risks and/or mitigation of DASE projects. 21% of the papers evaluated a practice, 9% evaluated a tool, 3% evaluated a method, and 2% evaluated a framework.

Table 21 presents the number and percentage of papers categorized by their context description. By combining the four context properties of research method used, context, subjects, and scale of the evaluations, the degree of realism of the studies can be found.

Table 21 The Context of the Data Collection Methods

Data Collection Method	*Context*	*Subjects*	*Research Evaluation*	*Papers*	*Percentage*
Observation	Industry	Employees	No	20	32%
Interviews	Industry	Employees	No	15	24%
Interviews	Industry	Employees	Practice	4	6.5%
Observation	Academic	Students	Tool	4	6.5%
Observation	Industry	Employees	Practice	3	4.5%

Observation and Interviews	Industry	Employees	No	3	4.5%
Observation	Academic	Students	Practice	2	3.5%
Research and Documentation	N/A	N/A	Practice	2	3.5%
Not Mentioned	N/A	N/A	Practice	1	1.5%
Not Mentioned	Industry	Employees	Method	1	1.5%
Experience	Industry	Employees	Practice	1	1.5%
Observation	Academic	Students	No	1	1.5%
Observation	Industry	Employees	Framework	1	1.5%
Observation	Industry	Employees	Method	1	1.5%
Observation	Industry	Employees	Tool	1	1.5%
Observation and Interviews	Academic	Students	Tool	1	1.5%
Research and Documentation	N/A	N/A	No	1	1.5%
Survey	Industry	Employees	No	1	1.5%

The distribution in Table 21 suggests that:

- It is noted that 20 papers (32%) had used Observation on Employees in an Industrial setting as part of their research. Additionally, 15 papers (24%) had used Interviews of Employees in an Industrial setting to conduct research.
- It is noted that Observation was most likely used in an Industry setting (using Employee subjects) than others.
- It is evident that in 81% of papers, employees were used as subjects in an Industrial setting while 13% were created in an academic setting.
- DASE practices were analyzed the most with 8 papers (13%) being evaluated in an Industry setting.

Based on this distribution in Table 21, it is fair to conclude that this review found a high degree of realism in the reviewed papers. A majority of challenges and workarounds were captured in an Industry setting using employee subjects. However, we do not have enough Industry evaluated papers that analyze practices, tools, methods, or frameworks. Methods used to approach DASE were analyzed in two papers, both in an Industry setting. Tools and Frameworks were both evaluated in an Industrial setting in one paper each. Those who wish to adopt DASE would be pleased with the trend of higher Industry based research. However, 35% of papers evaluate a method, practice, framework, or tool while 65% captured risks and/or mitigations. To further breakdown, 25% of the papers evaluated a practice, tool, framework, or method in an Industry setting. Since the goal of this paper was not to capture best practices in DASE, lower coverage of practices, tools, methods, and frameworks does not lower the degree of realism of this study.

7. Recommendations

Based on our observations of the reviewed publications in the area of distributed agile software engineering, we find that although a wealth of strong evidence is already provided for DASE, there are still a number of issues that needs to be addressed by practitioners and researchers when reporting on their experience with DASE. Covering these aspects when reporting on experience

with DASE would enable the community to draw stronger and deeper conclusions about the success or failure of projects adopting DASE. We believe addressing the following issues when reporting experience could significantly help improve the current state of the art in DASE:

- *Documenting Success and Failure*: When it comes to documenting success and failure in DASE projects, it was noted that none of the 63 papers had reported failure in their project. More importantly, 16 papers did not report either success or failure. In order to understand which model has a higher success ratio, researchers need to document exactly which model was used and whether their project was a success or not. By documenting failed projects, analysis could be done on the issues faced and any failed mitigation strategies followed to overcome challenges. Failure can also be seen as cases where deviation happened for various reasons from the initial set out plan. For instance, researchers and practitioners can report on whether their approach met its expected objectives as initially set out or not. If deviations or alterations had to be made, what were those and to what extent did that change the initial plans. Additionally, it helps understand the variables that directly affect success or failure (in both senses) – such as CMMI level, Infrastructure, resource models, and experience. It seems that the current literature is now more inclined towards reporting success in the deployment of DASE. While this is very beneficial, reporting on failure can also provide deep insight as to what needs to be considered or avoided when planning for DASE.
- *Criteria for success and their measurement*: Although 35 papers had reported that they were successful, criteria of success were not defined in most of these papers. Success criteria vary between organizations - budget, quality, and time to market. Since none of the papers had explicitly reported a failed project, it could be assumed that projects engaging in DASE have achieved cost and time to market benefits. Additionally, as mentioned above, organizational processes and resource models also have an impact on success. Therefore, it is important that reports on the success of DASE include the criteria for determining the success of the project. In other words, what were the criteria that were used to determine that the project was a success. In addition to the criteria, unambiguous and repeatable measurement mechanisms need to be reported so that similar studies could be replicated later for the sake of comparison.
- *Experience of human resources*: 37% of papers did not report on the level of experience of human resources in DASE, 33% of papers did not report on the level of experience working with distributed resources, and 35% of papers did not report on the level of experience working with Agile methods. It is safe to assume that past experience working with a model helps bring success to future projects. For this reason, it would be beneficial if the experience of the subjects involved in the experiments or actual deployment scenario is also reported. There may be situation when the DASE adoption strategy is strong but the project fails as a result inexperienced human resources. The community would need to be able to distinguish between the reasons that pertain specifically to

DASE adoption and secondary factors such as experience of human resources in work under DASE conditions.

- *Peak time zone difference*: 40% of the papers did not report any information about the peak time zone difference between distributed teams. Understanding time difference between teams is important given the fact that our review showed that synchronous communication is among the best workarounds when dealing with communication issues (12%) closely followed by formal and informal communication (10%) and overlapping work timings between teams (5%). Therefore, peak time zone difference can have significant impact on the success of DASE. For instance, two projects adopting exactly the same form of DASE but only with a different peak time zone in their teams can end up with different success or failure stories. This highlights the importance of carefully reporting peak time zone differences in the DASE setting.
- *Collocation of teams*: Collocation helps build trust and improves working relationship as the project progresses. Collocation was deemed as the second best workaround when dealing with challenges (10%). While collocation of teams shows impact on the success of DASE, not all reports included specific information on how collocation was achieved in their work. From among the reports that did mention whether collocation was used during the project, many did not report on the type of collocation model that was used (seed visits, maintaining visits, or both). It would be important to understand whether DASE projects engaged in collocation and what types of collocation in order to draw valid conclusion regarding their impact on the success or failure of DASE. For instance, this information could assist in answering questions such as whether seed visits are enough to build trust and relationships or not.

We would like to point out that not all of these specified data are pertinent or relevant to all studies in DASE; therefore, reports need to only cover the above aspects as much as they relate to the objectives of their study.

8. Direction for Future Work

Based on the review conducted, future work could be conducted on the following topics:

1. In our study, it was noted that there needed to be more tracking of success and failure of DASE projects. Some of the variables that could directly affect success of a DASE project include:
    a. Agile method: Agile methods have variations in the way human resources collaborate. As an example, Pair Programming requires human resources to share a desktop during the development and unit-testing phases. Pair programming can be done with distributed human resources using collaboration tools.

b. Success and failure criteria: Success and failure criteria vary between projects and organizations. Documenting and understanding criteria that define a projects success is important as it can help understand the overall success rate between Agile methods.
c. Experience of human resources in Agile and working with distributed resources: Past experience working with Agile methods and working with distributed resources can improve chances of success on future DASE initiatives.
d. Collocation strategy: Agile methods prefer more collocated collaborations. Since such is not feasible with distributed human resources, the human resources try to maintain seed visits and/or regular visits over the course of the project.
e. Time-zone difference: Time-zone differences can make a difference between teams communicating synchronously or not. Time difference of more than 5 hours could increase coordination and communication challenges.

By using these variables to perform a survey within current organizations that engage in DASE, practitioners could get a better understanding of what needs to be done prior to engaging in DASE.

2. Frameworks, practices, tools, and methods tend to incorporate the best of each category in order to assist projects. In our study, it was noted that 21% had experimented on a practice, 9% using tools, 3% using methods, and 2% using frameworks. It was clearly evident based on this review that Agile methods need to be tailored when working with distributed resources. The tailoring process could be vast involving several combinations of frameworks, practices, tools, and methods. Each organization tends to tailor models in their own ways based on their past experiences. By interviewing practitioners and integrating best methods and practices, future practitioners can use proven ways to implement DASE projects. Additionally, there are a vast number of frameworks and tools available that attempt to solve DASE issues – frameworks and tools for distributed story capture, development collaboration, and tracking quality assurance. A study could be conducted by experimenting between various frameworks and tools to better understand what works best in DASE under different circumstances.
3. In our study, it was noted that tailored methodologies, such as ScRumUp, lean development and Scrum, and Scrum and XP explicitly reported higher success compared to non-tailored methodologies [6]. Ways in which these methods were tailored was not described. Further studies could be conducted on tailored methodologies to see if fewer challenges are faced when working with tailored methodologies using distributed resources. Additionally, as mentioned above, success and failure criteria could be well defined to assess the outcome of tailored methodologies.
4. A majority of issues reported in this study fell under the communication and collaboration category. This is due to the fact that Agile methods require higher level of coordination and communication while both coordination and communication are the most difficult when working with distributed human resources. Architecture centric design proposed

ways to minimize or remove dependencies between teams thereby decreasing the amount of coordination and communication. In our study, we found architecture centric and modularized development listed 8 times (3%) as being a workaround to collaboration related challenges. However, only one paper had performed an experiment to assess success in DASE [32]. Detailed studies of how architectural centric design could benefit DASE projects could be conducted to assess impact on collaboration and communication risks.

5. The focus of this study was to better understand the DASE area. Since very little was captured in regards to the frameworks, practices, tools, and methods, future systematic literature review could be conducted on gathering frameworks, practices, tools, and methods in DASE. Such studies do not have to be limited to Agile projects as results from non-Agile distributed project could help Agile projects.

APPENDIX A

Primary Studies and the Results Summary

Paper # in Reference List	Study	Phase	Research Method	Agile Method	Quality
[1]	How agile practices have been tailored for adoption to distributed development and supporting GSD practices employed.	All Phases	Interviews	Scrum	5.0
[2]	Understand how DASE works on large projects.	All Phases	Interviews	Scrum	12.0
[3]	Do Scrum practices provide any advantages over traditional software engineering methods when used in globally distributed projects?	All Phases	Interviews	Scrum	12.0
[4]	Isolate and focus on the role of auditors, tools, and testers on distributed projects using Agile and Scrum.	QA	Observation	Scrum	5.0
[5]	Understand how DASE performs when executed in overall SDLC	All Phases	Observation and Interviews	Scrum	6.5
[6]	To understand if a new custom methodology, ScrumUP, is analysis for feasibility in GSD	All Phases	Observation	scRumUP	8.0
[7]	To understand if success can be constantly achieved in DASE	All Phases	Observation	Scrum and XP	7.0
[8]	How well do Agile and Scrum practices support distributed teams using Tools to implement.	All Phases	Observation	Scrum	10.5
[9]	Describes the experience of two large globally distributed companies implementing Scrum.	All Phases	Observation	Scrum	6.5
[10]	How does DASE work on a long term project?	All Phases	Interviews	Scrum	7.5
[11]	Can a team establish a localized velocity and quality and then maintain or increase that velocity and quality when distributing teams across continents	All Phases	Observation	Scrum and XP	5.0
[12]	Discusses the advantages and challenges of combining GSE with agile development based on a theoretical based research.	All Phases	Research and Documentation	Scrum and XP	3.5
[13]	What are the best practices when adopting DASE?	All Phases	Observation	Scrum	4.0
[14]	To understand if the medium and method of communication effects success in DASE	All Phases	Observation and Interviews	Agile, specifics not mentioned	5.0
[15]	Proposes a holistic approach that supports management of the development progress in geographically distributed agile projects by identifying and co-coordinating the impact of the technical factors on progress.	All Phases	Interviews	XP	3.0
[16]	Captures the experience of a vendor house in handling Distributed Agile projects. Discusses a validated model to make a smooth transition from a collocated to a distributed scenario in agile projects.	All Phases	Experience	XP	8.0
[17]	Describes experience of key resources in a large DASE project. Explains how a team was able to make changes that allowed the continuous conversations to take place.	Development	Observation	XP	8.0
[18]	Paper proposes FLOW Mapping, a systematic approach for planning and managing information flows in distributed projects.	All Phases	Observation	XP	13.5

[19]	Describes how the Agile coaches can help team members adopt agile practices.	All Phases	Observation	Scrum	7.5
[20]	Using experience from two globally distributed outsourcing partners, this paper analysis a comprehensive test automation strategy for their agile teams that effectively leveraged both in house and outsourced activities.	QA	Observation	Scrum	9.0
[21]	This study aims to develop a new framework to identify the dynamic risks in GDSD projects and mitigate them using agile risk management practices.	All Phases	Research and Documentation	Agile, specifics not mentioned	6.0
[22]	Paper conducts a rapid yet intensive Agile Crash Course (on job learning by doing with full-time support by skilled coaches) based on principles of Lean Software Development.	All Phases	Research and Documentation	Agile, specifics not mentioned	12.0
[23]	The paper reports a multi-case study that investigates the impact of key project contextual factors on the use of Scrum practices in GSD.	All Phases	Observation	Scrum	9.0
[24]	"Papers shows best practices when dealing with the lack of communication in a distributed Scrum Team.	All Phases	Observation	Scrum	7.0
[25]	Paper reviews a distributed card based planning tool (since physical cards are not available in DASE)	Requirement	Observation	Agile, specifics not mentioned	12.5
[26]	Best practices in collaboration that can improve success in DASE	All Phases	Observation	Scrum and XP	10.5
[27]	Explains how trust can determines the success or failure of distributed Agile projects, and describes how trust can be generated and sustained by increasing effective communication and understanding cultural differences.	All Phases	Observation	Agile, specifics not mentioned	10.0
[28]	Shows how communication challenges can be tackled with common guiding and design metaphors, architecture-centric development, task assignments with component tasks and extensive quality assurance measures.	All Phases	Observation	XP	4.0
[29]	This paper attempts to understand communication in DASE	All Phases	Observation	Agile, specifics not mentioned	10.0
[30]	Proposes a process that adds a level of governance to improve success in DASE	All Phases	Observation	XP	10.0
[31]	Conducts a study to see if Agile can be adopted on teams that do not share programming responsibility	All Phases	"Survey		11.0
[32]	Performs research in which the relation between large scale projects and agile approaches to software development is studied.	All Phases	Observation	Agile, specifics not mentioned	9.0
[33]	Documents best practices to improve coordination in distributed Agile projects.	All Phases	Observation	XP	10.0
[34]	Conducts a detailed study of a software development organization following Scrum for developing software products using distributed resources.	All Phases	Observation	Scrum	7.0
[35]	Documents best practices in DASE	All Phases	N/A	Scrum	3.5
[36]	Documents potential area of issues when engaging in Agile using distributed resources	All Phases	Interviews	Agile, specifics not mentioned	9.5
[37]	Documents strategies that Project Managers can use when working on Agile projects using distributed resources.	All Phases	Interviews	Agile, specifics not mentioned	11.0
[38]	Shows how planning can be improved in DASE	Planning	Interviews	Scrum	9.5
[39]	Effects of culture, competence, and knowledge asymmetry in DASE.	All Phases	Observation and Interviews	Scrum and XP	6.0
[40]	Construct a preliminary conceptual model for exploring three proposed dimensions necessary for successful configuration of global agile teams: structure, agility, and virtual-ness.	All Phases	Observation and Interviews	Scrum and XP	7.0
[41]	Proposes a method of having semi self-organized teams as being a promising motivating factor in DASE	All Phases	Observation	Agile, specifics not[mentioned	4.0
[42]	Explains the issues faced when going from Waterfall to Agile on a large distributed projects	All Phases	Observation	Lean development and Scrum	10.0
[43]	Paper shows how computational, coordination, organizational, distributional, and communicational models offers a high degree of flexibility regarding architectural and design changes.	Design	Not mentioned	Agile, specifics not mentioned	5.0
[44]	Documents lessons learned from projects that have implemented Agile using distributed resources.	All Phases	Interviews	Agile, specifics not mentioned	9.0
[45]	Highlighting challenges and success during My Yahoo development - DASE.	All Phases	Observation	Scrum	9.5
[46]	Shows the importance of tools when developing a product using Pair Programming using distributed resources.	Development	Observation	Pair Programmin	6.0

				g	
[47]	Documents key concerns in DASE	All Phases	Interviews	Scrum	9.5
[48]	Provides guidelines that organizations can follow in DASE	All Phases	Interviews	Scrum	10.0
[49]	Paper outline some of the strategies and challenges associated with implementing agile methods in distributed software project teams.	All Phases	Observation	Agile, specifics not mentioned	5.0
[50]	This paper describes a technique called Silent Grouping that can be used to compliment Planning Poker, explaining how to apply it so that large sets of user stories can be sized in minutes.	Planning	Observation	Agile, specifics not mentioned	9.0
[51]	Evaluates tools that could be used for planning in DASE.	Planning	Observation	Agile, specifics not mentioned	5.0
[52]	Challenges that organizations can face when going from Waterfall to Agile using distributed resources.	All Phases	Interviews	Agile, specifics not mentioned	7.0
[53]	This study presents a framework that integrates best practices of adapting and applying agile methods reported in the literature	All Phases	Observation	Scrum	11.5
[54]	Paper investigate how agile teams can be distributed by adding a "remote partner" – and still maintain agile advantages.	All Phases	Observation	XP	6.5
[55]	Provides a unique view from the point of view of Agile "expert" practitioners on the use of Agile using distributed resources.	All Phases	Interviews	Scrum and XP	11.5
[56]	Experience of a global multinational company on transitioning from distributed and traditional to distributed and agile.	All Phases	Interviews	Scrum	8.0
[57]	How to work with vendors/subcontractors in DASE	All Phases	Interviews	Scrum	10.0
[58]	Paper outlines some of the typical challenges that could be met during real-world commercial projects, and how they could be solved.	All Phases	Observation	Scrum	6.0
[59]	How to work with a vendor in DASE where the vendor has a higher CMMI level	All Phases	Interviews	Scrum	7.0
[60]	Paper describes how scrum practices could be successfully applied in a distributed setting	All Phases	Interviews	Scrum and XP	12.0
[61]	Explains how a typical DASE project can face particular control challenges related to balancing fixed vs. evolving quality requirements and people vs. process-based collaboration.	All Phases	Interviews	Scrum and XP	11.0
[62]	Paper analysis agile software development literature by analyzing decisions made during the iteration cycle and identifying six key obstacles to these decisions	All Phases	Interviews	Scrum	10.0
[63]	Paper analyses the structure and use of story cards and the Wall in three mature XP teams, using a distributed cognition approach.	Planning	Observation	XP	9.0

APPENDIX B

Quality Assessment

Criteria	Yes	No	Somewhat
Problem defined?	97%	0%	3%
Research Questions?	38%	62%	0%
Domain of evaluation?	100%	0%	0%
Samples/instruments used in research?	75%	25%	0%
Data Collection explained?	54%	24%	2%
Data Analysis explained?	52%	48%	0%
Interpretation of analysis?	63%	37%	0%
Results explained in detail?	96%	4%	0%
Assumptions described?	25%	71%	4%
Threats described?	10%	87%	3%
Lessons Learned?	38%	60%	2%
Practical Implications explained?	97%	3%	0%
Related Work explained?	30%	70%	0%
Recommendation for future work?	35%	52%	13%

APPENDIX C

Data Extraction Properties

#	Property	Values	Paper Selection	RQ Mappings
1	1.1 Is the context of the study described? 1.2 What is the problem? 1.3 Where does it occur?	The context provided in the reviewed literature	Introduction	RQ 1.1, RQ 1.2
2	2.1 Has the paper provided reasons for engaging in DASE? 2.2 What phase of the project lifecycle has utilized distributed human resources? 2.3 What is the human resource distribution model? 2.4 Do human resources have experience in distributed Agile software engineering? 2.5 Do human resources have experience in distributed teams? 2.6 Do human resources have experience in Agile? 2.7 Have human resources been collocated at some point during the project lifecycle?	2.1 – [Business Practice, Experiment, Heard of Agile, Not Mentioned, To Simulate] 2.2 – [All Phases, Design, Development, Planning, QA and Testing, Requirements] 2.3 – Number of teams and how far are they located 2.4 – [Yes, No, Some Resources, Not Mentioned] 2.5 – [Yes, No, Some Resources, Not Mentioned] 2.6 – [Yes, No, Some Resources, Not Mentioned] 2.7 - – [Seed Visits, Maintenance Visits, Seed and Maintenance Visits, No, Not Mentioned]	Background	RQ 1.1, RQ 1.2, RQ 1.3
3	3.1 What type of research method has been used in this study? 3.2 In which environment has this study taken place? 3.3 Who are the subjects of this research? 3.4 Does the paper evaluate a practice, method, tool, or framework?	3.1 – [Experience, Interview, Not Mentioned, Observation, Observation and Interviews, Research and Documentation, Survey] 3.2 – [Industry, Academic, Not Mentioned] 3.3 – [Employees, Students, Not Mentioned] 3.4 – [Practice, Method, Tool, Framework, Capture Risks/Mitigation]	Research Method	RQ 4.1, RQ 4.2
4	4.1 Are risks and issues documented? 4.2 Are workaround or mitigation plans listed? 4.3 What type of Agile model was used? 4.4 Was there a specific reason to use an Agile mode?	4.1 - A list of challenges, risks, and issues documented 4.2 – A list of solutions, workaround, and mitigation plan 4.3 – [Agile – specifics not mentioned, Scrum, XP, Scrum and XP, Lean Development and Scrum, Pair Programming, ScRumUp] 4.4 – [Business Practice, Experience, Experiment, Knowledge within the team, Research, Not Mentioned]	Results	RQ 2.1, RQ 2.2, RQ 3.1
5	5.1 Was the project considered a success? 5.2 Will the organization engage in DASE?	5.1 – [Yes, No, Some what, Not Mentioned] 5.2 – [Yes, No, Not Mentioned]	Conclusion	RQ 3.1, RQ 3.2

REFERENCES

[1] Paasivaara, M.; Durasiewicz, S.; Lassenius, C., “Using Scrum in Distributed Agile Development: A Multiple Case Study” ICGSE 2009. Fourth IEEE International Conference on Global Software Engineering, 2009, 2009, 10.1109/ICGSE.2009.27

[2] Paasivaara, M.; Durasiewicz, S.; Lassenius, C., “Distributed Agile Development: Using Scrum in a Large Project” ICGSE 2008. IEEE International Conference on Global Software Engineering, 2009, 2008, 10.1109/ICGSE.2008.38

[3] Bannerman, P.L.; Hossain, E.; Jeffery, R., "Scrum Practice Mitigation of Global Software Development Coordination Challenges: A Distinctive Advantage?" System Science (HICSS), 2012 45th Hawaii International Conference on Global Software Engineering, 2009, 2012, 10.1109/HICSS.2012.512

[4] Scharff, C., “Guiding global software development projects using Scrum and Agile with quality assurance” Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on, 2011, 10.1109/CSEET.2011.5876097 "

[5] Su, S.H.; Scharff, C., “Know Yourself and Beyond: A students' global software development project experience with Agile Methodology” 2010 5th International Conference on Global Software Engineering, 2009, 2010, 10.1109/ICCSE.2010.5593595 "

[6] del Nuevo, E.; Piattini, M.; Pino, F.J., “Scrum-based Methodology for Distributed Software Development” Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on Global Software Engineering, 2009, 2011, 10.1109/ICGSE.2011.23 "

[7] Sutherland, J.; Schoonheim, G.; Rustenburg, E.; Rijk, M., “Fully Distributed Scrum: The Secret Sauce for Hyperproductive Offshored Development Teams” GILE '08. Conference Global Software Engineering, 2009, 2008, 10.1109/Agile.2008.92 "

[8] Scharff, C.; Gotel, O.; Kulkarni, V., “Transitioning to Distributed Development in Students' Global Software Development Projects: The Role of Agile Methodologies and End-to-End Tooling” 2010 Fifth International Conference on Global Software Engineering, 2009, 2010, 10.1109/ICSEA.2010.66

[9] Cristal, M.; Wildt, D.; Prikladnicki, R., “Usage of SCRUM Practices within a Global Company” ICGSE 2008. IEEE International Conference on Global Software Engineering, 2009, 2008, 10.1109/ICGSE.2008.34

[10] Paasivaara, M.; Lassenius, C., “Scaling Scrum in a Large Distributed Project” 2011 International Symposium on Global Software Engineering, 2009, 2011, 10.1109/ESEM.2011.49

[11] Sutherland, J.; Schoonheim, G.; Rijk, M., “Fully Distributed Scrum: Replicating Local Productivity and Quality with Offshore Teams” HICSS '09. 42nd Hawaii International Conference on Global Software Engineering, 2009, 2009, 10.1109/HICSS.2009.225

[12] Dullemond, K.; van Gameren, B.; van Solingen, R., “How Technological Support Can Enable Advantages of Agile Software Development in a GSE Setting” ICGSE 2009. Fourth IEEE International Conference on Global Software Engineering, 2009, 2009, 10.1109/ICGSE.2009.22

[13] Hansen, M.T.; Baggesen, H., “From CMMI and Isolation to Scrum, Agile, Lean and Collaboration” Agile Conference, 2009. AGILE '09., 2009, 10.1109/AGILE.2009.18

[14] Niinimaki, T., “Face-to-Face, Email and Instant Messaging in Distributed Agile Software Development Project” 2011 Sixth IEEE International Conference on Global Software Engineering, 2009, 2011, 10.1109/ICGSE-W.2011.15

[15] Alyahya, S.; Ivins, W.K.; Gray, W.A., “Co-ordination Support for Managing Progress of Distributed Agile Projects” 2011 Sixth IEEE International Conference on Global Software Engineering, 2009, 2011, 10.1109/ICGSE-W.2011.24 "

[16] Sureshchandra, K.; Shrinivasavadhani, J., “Adopting Agile in Distributed Development” ICGSE 2008. IEEE International Conference on Global Software Engineering, 2009, 2008, 10.1109/ICGSE.2008.25 "

[17] Williams, W.; Stout, M., "Colossal, Scattered, and Chaotic (Planning with a Large Distributed Team)" Agile, 2008. AGILE '08. Conference, 2008, 10.1109/Agile.2008.25

[18] Stapel, K.; Knauss, E.; Schneider, K.; Zazworka, N., “FLOW Mapping: Planning and Managing Communication in Distributed Teams” 2011 6th IEEE International Conference on Global Software Engineering, 2009, 2011, 10.1109/ICGSE.2011.9 "

[19] Paasivaara, M., “Coaching Global Software Development Projects” 2011 6th IEEE International Conference on Global Software Engineering, 2009, 2011, 10.1109/ICGSE.2011.33 "

[20] Little, Todd; Elliott, Suzanne; Hughes, Joe; Simion, Florin, "Leveraging Global Talent for Effective Test Agility" Agile Conference (AGILE), 2012, 2012, 10.1109/Agile.2012.25

[21] Mudumba, V.; One-Ki Lee, “A New Perspective on GDSD Risk Management: Agile Risk Management” 2010 5th IEEE International Conference on Global Software Engineering, 2009, 2010, 10.1109/ICGSE.2010.33 "

[22] Prochazka, J.; Kokott, M.; Chmelar, M.; Krchnak, J., “Keeping the Spin -- From Idea to Cash in 6 Weeks: Success Story of Agile/Lean Transformation” 2011 6th IEEE International Conference on Global Software Engineering, 2009, 2011, 10.1109/ICGSE.2011.19"

[23] Hossain, E.; Bannerman P.;, Jeffery, R., "Towards an understanding of tailoring scrum in global software development: a multi-case study" ICSSP '11: Proceedings of the 2011 International Conference on Software and Systems Process, 2011, 10.1145/1987875.1987894

[24] Bless, M., "Distributed Meetings in Distributed Teams" Lecture Notes in Business Information Processing, 2010, Volume 48, Part 3, 251-260,, 2010, 10.1007/978-3-642-13054-0_27

[25] Morgan, R.; Maurer, F.; Chiasson, M., "An Observational Study of a Distributed Card Based Planning Environment" Lecture Notes in Business Information Processing, 1, Volume 9, Agile Processes in Software Engineering and Extreme, 2008, 10.1007/978-3-540-68255-4_6

[26] Sharp, H.; Giuffrida R.; and Melnik, G., "Information Flow within a Dispersed Agile Team: A Distributed Cognition Perspective" Lecture Notes in Business Information Processing, 1, Volume 111, Agile Processes in Software Engineering and Extreme Programming, Part 2, Pages 62-76, 2012, 10.1007/978-3-642-30350-0_5

[27] Dorairaj, S.; Noble, J.; Malik, P., "Understanding the Importance of Trust in Distributed Agile Projects: A Practical Perspective" Lecture Notes in Business Information Processing, 1, Volume 48, Agile Processes in Software Engineering and Extreme Programming, Part 2, Pages 172-177, 2010, 10.1007/978-3-642-13054-0_14

[28] Kornstädt, A.; and Sauer, J., “Mastering Dual-Shore Development: The Tools and Materials Approach Adapted to Agile Offshoring” Lecture Notes in Computer Science, 2007, Volume 4716, Software Engineering Approaches for Offshore and Outsourced Development, Pages 83-95, 2007, 10.1007/978-3-540-75542-5_7

[29] Korkala, M.; Pikkarainen, M.; Conboy, K., "Distributed Agile Development: A Case Study of Customer Communication Challenges" Lecture Notes in Business Information Processing, 1, Volume 31, Agile Processes in Software Engineering and Extreme Programming, Part 3, Part 5, Pages 161-167, 2009, 10.1007/978-3-642-01853-4_21

[30] Klein, H.; Knauss, K.; and Rausch, A, “Scaling Software Development Methods from Co-located to Distributed” Lecture Notes in Business Information Processing, 1, Volume 94, Software Quality. Process Automation in Software Development, Part 3, Pages 71-83, 2012, 10.1007/978-3-642-27213-4_6

[31] Korhonen, K., "Adopting Agile Practices in Teams with No Direct Programming Responsibility: A Case Study" Lecture Notes in Computer Science, 2011, Volume 6759, Product-Focused Software Process Improvement, Pages 30-43, 2011, 10.1007/978-3-642-21843-9_5

[32] Bosch, J.; Bosch-Sijtsema, P., "Coordination Between Global Agile Teams: From Process to Architecture" Agility Across Time and Space, 2010, Part 3, 217-233, 2010, 10.1007/978-3-642-12442-6_15

[33] Hole, S.; and Moe, NB., "A Case Study of Coordination in Distributed Agile Software Development" Communications in Computer and Information Science, 2008, Volume 16, Software Process Improvement, Part 5, Pages 189-200, 2008, 10.1007/978-3-540-85936-9_17

[34] Inayat, I.; Muhammad Asim Noor, MA.; Inayat, Z., "Facilitating an Off-Site Customer in Product-Based Agile Software Development: An Industrial Case Study" Communications in Computer and Information Science, 2012, Volume 281, 210-221, 2012, 10.1007/978-3-642-28962-0_21

[35] Šmite, D.; Moe, NB.; Ågerfalk, PJ., "Fundamentals of Agile Distributed Software Development" Agility Across Time and Space, 2010, Part 1, 3-7,, 2010, 10.1007/978-3-642-12442-6_1

[36] Srinivasan, J., "Preparing your Offshore Organization for Agility: Experiences in India" Agility Across Time and Space, 2010, Part 2, Pages 117-130, 2010, 10.1007/978-3-642-12442-6_8

[37] Hossain, E.; Babar, MA.; and Verner, J., "Towards a Framework for Using Agile Approaches in Global Software Development" Lecture Notes in Business Information Processing, 2009, Volume 32, Part 4, 126-140, 2009, 10.1007/978-3-642-02152-7_11

[38] Svejvig, P.; Nielsen, AF., "The Dilemma of High Level Planning in Distributed Agile Software Projects: An Action Research Study in a Danish Bank" Agility Across Time and Space, 2010, Part 3, Pages 171-182, 2010, 10.1007/978-3-642-12442-6_12

[39] Pries-Heje, J.;, Hansen, M.; Knudsen, SB., "When Global Process Fails: A Grounded Theory Study of a Case from Agile Engagement to Compulsive Outsourcing" IFIP Advances in Information and Communication Technology, 2010, Volume 334, E-Government, E-Services and Global Processes, Pages 245-258, 2010, 10.1007/978-3-642-15346-4_20

[40] Sharp, JH.; Ryan, SD., "A Preliminary Conceptual Model for Exploring Global Agile Teams" Lecture Notes in Business Information Processing, 2008, Volume 9, Part 6, 147-160, 2008, 10.1007/978-3-540-68255-4_15

[41] Kumlander, D., "Motivating Company Personnel by Applying the Semi-self-organized Teams Principle" Innovations in Computing Sciences and Software Engineering, 2010, Pages 245-248, 2010, 10.1007/978-90-481-9112-3_41

[42] Schnitter, J.; Mackert, O., "Large-Scale Agile Software Development at SAP AG" Communications in Computer and Information Science, 2011, Volume 230, 209-220, 2011, 10.1007/978-3-642-23391-3_15

[43] Mordinyi, R.; Kühn, E.; Schatten, A., "Structuring Complexity Issues for Efficient Realization of Agile Business Requirements in Distributed Environments" Lecture Notes in Business Information Processing, 2010, Volume 48, Part 2, 202-207, 2010, 10.1007/978-3-642-13054-0_19

[44] Paasivaara, M.; Lassenius, C., "Using Scrum Practices in GSD Projects" Agility Across Time and Space, 2010, Part 4, 259-278,, 2010, 10.1007/978-3-642-12442-6_17

[45] Lee, S.; Yong, HS., "Distributed agile: project management in a global environment" Empirical Software Engineering, 2010, Volume 15, Number 2, 2010, 10.1007/s10664-009-9119-7

[46] Dajda, J.; Dobrowolski, G., "How to Build Support for Distributed Pair Programming" Lecture Notes in Computer Science, 2007, Volume 4536/2007, 70-73, 2007, 10.1007/978-3-540-73101-6_10

[47] Dorairaj, S.; Noble, J.; Malik, P., "Understanding Team Dynamics in Distributed Agile Software Development" Lecture Notes in Business Information Processing, 2012, Volume 111, Agile Processes in Software Engineering and Extreme Programming, Part 2, Pages 47-61, 2012, 10.1007/978-3-642-30350-0_4

[48] Korkala, M.; Pikkarainen, M.; Conboy, K., "Combining Agile and Traditional: Customer Communication in Distributed Environment" Agility Across Time and Space, 2010 Part 3, Pages 201-216, 2010, 10.1007/978-3-642-12442-6_14

[49] Maruping, LM., "Implementing Extreme Programming in Distributed Software Project Teams: Strategies and Challenges" Agility Across Time and Space, 2010, Part 2, Pages 11-30, 2010, 10.1007/978-3-642-12442-6_2

[50] Power, K., "Using Silent Grouping to Size User Stories" Lecture Notes in Business Information Processing, 2011, Volume 77, Part 1, 60-72, 2011, 10.1007/978-3-642-20677-1_5

[51] Wang, X.; Maurer, F.; Morgan, R.; Oliveira, J., "Tools for Supporting Distributed Agile Project Planning" Agility Across Time and Space, 2010, Part 3, Pages 183-199, 2010, 10.1007/978-3-642-12442-6_13

[52] Tureček, T.; Šmiřák, R.; Malík, T.; Boháček, P., "Energy Project Story: From Waterfall to Distributed Agile" Lecture Notes in Business Information Processing, 2010, Volume 48, Part 3, 362-371, 2010, 10.1007/978-3-642-13054-0_39

[53] van Hillegersberg, J.; Ligtenberg, G.; Aydin, MN., "Getting Agile Methods to Work for Cordys Global Software Product Development" Lecture Notes in Business Information Processing, 2011, Volume 91, Part 1, 133-152, 2011, 10.1007/978-3-642-24815-3_8

[54] Meyer, S.; Knauss, E.; Schneider, K., "Distributing a Lean Organization: Maintaining Communication While Staying Agile" Lecture Notes in Business Information Processing, 2010, Volume 65, Part 2, 99-103, 2010, 10.1007/978-3-642-16416-3_14

[55] Dorairaj, S.; Noble, J.; Malik, P., "Effective Communication in Distributed Agile Software Development Teams" Lecture Notes in Business Information Processing, 2011, Volume 77, Part 1, 102-116, 2011, 10.1007/978-3-642-20677-1_8

[56] Wildt, D.; Prikladnicki, R., "Transitioning from Distributed and Traditional to Distributed and Agile: An Experience Report" Agility Across Time and Space, 2010, Part 2, Pages 31-46, 2010, 10.1007/978-3-642-12442-6_3

[57] Rudzki, J.; Hammouda, I.; Mikkola, T.; Mustonen, K.; Systä, T., "Considering Subcontractors in Distributed Scrum Teams" Agility Across Time and Space, 2010, Part 3, Pages 235-255, 2010, 10.1007/978-3-642-12442-6_16

[58] Sadun, C., "Scrum and Global Delivery: Pitfalls and Lessons Learned" Agility Across Time and Space, 2010, Part 2, Pages 71-89, 2010, 10.1007/978-3-642-12442-6_5

[59] Kussmaul, C., "Onshore and Offshore Outsourcing with Agility: Lessons Learned" Agility Across Time and Space, 2010, Part 2, Pages 91-106, 2010, 10.1007/978-3-642-12442-6_6

[60] Paasivaara, M.; Durasiewicz,S.; Lassenius, C., "Using scrum in a globally distributed project: a case study" Software Process: Improvement and Practice, Volume 13, Issue 6, pages 527–544, 2008, 10.1002/spip.402

[61] Persson, JS.; LarsMathiassen, L.; Aaen, L., "Agile distributed software development: enacting control through media and context" Information Systems Journal, 2011, 10.1111/j.1365-2575.2011.00390.x

[62] Drurya, M.; Conboy,K.; Power, K., "Obstacles to decision making in Agile software development teams" Journal of Systems and SoftwareVolume 85, Issue 6, June 2012, Pages 1239–1254, 2012, 10.1016/j.jss.2012.01.058

[63] Sharp, H.; Robinson H., "Collaboration and co-ordination in mature eXtreme programming teams" International Journal of Human-Computer Studies archive, Volume 66 Issue 7, July, 2008, Pages 506-518, 2008, 10.1016/j.ijhcs.2007.10.004

[64] Fitzgerald, B. et al., "Customising Agile Methods to Software Practices at Intel Shannon" European Journal of Information Systems 15, pp. 200–213. 2006, DOI= 10.1057/palgrave.ejis.3000605

[65] Munch, J. et al., "Understanding Lacking Trust in Global Software Teams: A Multi-case Study" Product-Focused Software Process Improvement, pp. 20-34, 2007, DOI=http://dx.doi.org/10.1007/978-3-540-73460-4_6

[66] Zopf, S., "Success factors for globally distributed projects" Software Process: Improvement and Practice, pp. 355-359, 2009, DOI=http://dx.doi.org/10.1002/spip.423

[67] Kitchenham, BA.; "Procedures for Performing Systematic Reviews", tech. report SE0401, Dept. of Computer Science, Univ. of Keele, and tech. report 0400011T.1, Empirical Software Eng., National Information and Communications Technology Australia, 30 Aug. 2004.

[68] Kitchenham, BA.; Pfleeger, S:.; Pickard, LM.; Jones, PW.; Hoaglin, DC.; El Emam, K.; Rosenberg, K.; "Preliminary guidelines for empirical research in software engineering", IEEE Transactions on Software Engineering 28 (8) (2002) 721–734.

[69] Runeson, P.; and H'ost, M., "Guidelines for conducting and reporting case study research in software engineering". Empirical Software Engineering, 131–164, 2009.

[70] Jedlitschka, A., Ciolkowski, M. & Pfahl, D., "Reporting experiments in software engineering", in F. Shull, J. Singer & D. Sjøberg, eds, 'Guide to Advanced Empirical Software Engineering', 2008, Springer-Verlag, London, chapter 8.

[71] Jalali, S.; Wohlin, C.,"Agile Practices in Global Software Engineering - A Systematic Map", 2010 5th IEEE International Conference on Global Software Engineering (ICGSE), Pg: 45-54, 2010, DOI=10.1109/ICGSE.2010.14

[72] Smite, D.; Wohlin, C.; Gorschek, T.; Feldt, R.; “Empirical evidence in global software engineering: a systematic review”, Empirical Software Engineering, Volume 15, Number 1, 2010, Pg: 91-118, DOI= 10.1007/s10664-009-9123-y

[73] Jalali, S.; Wohlin, C., “Global software engineering and agile practices: a systematic review”, Journal of Software: Evolution and Process, Special Issue on Global Software Engineering, Volume 24, Issue 6, pg:643–659, 2012, DOI=10.1002/smr.561

[74] Mishra, D.; Mishra, A., “Problems and Solutions in Distributed Software Development: A Systematic Review”, Lecture Notes in Computer Science, Lecture Notes in Business Information Processing, 2009, Volume 16, 107-125, DOI=10.1007/978-3-642-01856-5_8

[75] Hossain, E.; Babar, M.A.; Paik, HY., “Using Scrum in Global Software Development: A Systematic Literature Review”, ICGSE 2009. Fourth IEEE International Conference on Global Software Engineering, 2009., Page(s): 175-184, DOI=10.1109/ICGSE.2009.25

[76] West, D.; Grant, T.; “Agile Development: Mainstream Adoption Has Changed Agility”, Forrester, 2010, http://pmshow2012.programmedevelopment.com/public/uploads/files/forrester_agile_development_mainstream_adoption_has_changed_agility.pdf

[77] Therrien,E., "Overcoming the Challenges of Building a Distributed Agile Organization", AGILE '08. Conference , pp. 368-372, 2008, DOI=10.1109/Agile.2008.9

[78] Young, C, Terashima H., "How Did We Adapt Agile Processes to Our Distributed Development?” Agile 2008, pp. 304-309, 2008, DOI=http://doi.ieeecomputersociety.org/10.1109/Agile.2008.7

[79] Smite, D. et al., “Empirical evidence of global software engineering: A systematic Review” Empirical Software Engineering, 2010, DOI= 10.1007/s10664-009-9123-y

[80] Dedrick, J.; Kraemer, K.L.; Dunkle, D.; “Offshoring of Software Development: Patters and Recession Effects”, Recent Work, Personal Computing Industry Center, University of California – Irvine, 2010, http://escholarship.org/uc/item/22r9q7wp

[81] Wright RW, Brand RA, Dunn W, Spindler KP. How to write a systematic review. Clin Orthop Relat Res 2007; 455:23-29.

[82] Biolchini, J., Gomes, P., Cruz, A., & Travassos, G. (2005). Systematic review in software engineering. Rio de Janeiro, Brazil: Systems Engineering and Computer Science Department, UFRJ.

[83] Maria Paasivaara and Casper Lassenius. 2006. Could Global Software Development Benefit from Agile Methods?. In Proceedings of the IEEE international conference on Global Software Engineering (ICGSE '06). IEEE Computer Society, Washington, DC, USA, 109-113.

[84] Estler, H.-C.; Nordio, M.; Furia, C.A.; Meyer, B.; Schneider, J., Agile vs. Structured Distributed Software Development: A Case Study, Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on , vol., no., pp.11,20, 27-30 Aug. 2012

[85] Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003.