Journal of Software: Evolution and Process

JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS
J. Softw. Evol. and Proc. 2015; **27**:723–762
Published online 8 June 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/smr.1718

A systematic review of distributed Agile software engineering

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

[1]*Athabasca University, Athabasca, Canada*
[2]*Ryerson University, Toronto, Canada*
[3]*The University of Edinburgh, Edinburgh, United Kingdom*

ABSTRACT

The combination of Agile methods and distributed software development via remote teams represents an emerging approach to address 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. 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 (DASE) practices. Second, we would like to investigate and find out the most important risks that threaten a DASE approach and what mitigation strategies exist to address them. Finally, we would like to highlight which of the available approaches among the existing Agile methodologies has 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.

We carried out a systematic literature review of DASE techniques and approaches. 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. Copyright © 2015 John Wiley & Sons, Ltd.

Received 9 January 2014; Revised 8 January 2015; Accepted 20 April 2015

KEY WORDS: distributed Agile software engineering; empirical results; surveys and field studies; Agile; 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, DSD is still evolving. As such, the failure rate of DSE projects is higher than collocated projects [1].

DSE allows the client organization to engage in activities across one or more remote sites [1]. 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 [1].

*Correspondence to: Ebrahim Bagheri, Ryerson University, Toronto, Canada.
†E-mail: bagheri@ryerson.ca

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 Blue Cross Blue Shield of Pennsylvania where EDS was responsible for managing Blue Cross's data [2]. 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 [2]. The commonality between these deals was 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 [2]. 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 [2]. 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 [3].

Agile methodologies are undertaken by organizations that are interested in delivering business value regularly and often [4]. 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 canceling projects and programs early on, therefore emphasizing on those projects that have a good return on investment [5] and [6]. 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 [7]. If the aforementioned statement is true, then XP projects can eliminate the need for extensive and formal requirements analysis and entire design of the system early on [8]. 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 [5, 6]. Even with these challenges, organizations wish to engage in DSE because of cost savings (40%), capacity of remote teams (20%), application knowledge (13%), and quick time to market (11%) [9]. Munch *et al*. [10] define 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 [11, 12]. However, the principles of DSE and Agile are not always necessarily

DOI: 10.1002/smr

compatible [13]. There is no collocation among team members resulting in no face-to-face interaction. Team cohesion does not exist because of cultural and language issues. Informal communication is also not certain depending on the time zones where sub-teams reside, which results in asynchronous communication. Projects with distributed human resources could require more documentation to avoid any issues around scope misunderstanding, which is not in-line with Agile characteristics [13].

In addition, XP practices such as pair programming are difficult to implement with distributed human resources. Agile revolves around informal practices and mutual adjustment, whereas DSE revolves around formal mechanisms and direct supervision [14]. Based on the aforementioned statement, it seems that Agile methods do not work with DSD projects. It cannot be the case that Agile practices be implemented partially as Agile insists that it should be implemented entirely in order to be efficient [15]. 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 [16, 17]. 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 among software engineers [18]. Since its inception in 2004, systematic literature reviews have gained significant popularity among software engineers [18]. 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 [18]. 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 DSD/engineering. The paper by Smite *et al.* [19] has focused on reviewing empirical evidence in global software engineering; therefore, it is not focused on aspects of Agile methodologies. Similarly, Jimenez and Piattini [20] focus on some major problems within the DSD domain in general and provide an overview of some of the suggested solutions. The work by Hossain *et al.* [24] is closer to the theme of our work and focuses on the review of the role and impact of the Scrum approach on DSD. However, it is only restricted to Scrum from the range of Agile methodologies. The works by Jalali and Wohlin [21, 22] 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.

DOI: 10.1002/smr

While the work by Jalali and Wohlin [21, 22] and 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 DASE practices. Second, we would like to investigate and find out the most important risks that can threaten a DASE approach and what mitigation strategies exist to address them. Finally, we 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 [21, 22] 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 [21, 22].

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 aforementioned questions*?
This question will clarify the degree of strength of evidence that is available within the literature to support the findings in this systematic literature review.

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

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.

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 the 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 performed 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 were taken into consideration.

DOI: 10.1002/smr

Table I. Designated research questions for the study.

Number	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 systems development life cycle (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?	Provided 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 work-arounds 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 aforementioned 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

DASE, distributed Agile software engineering.

Search query in Table II was used on Google Scholar and the Digital Bibliography & Library Project Computer Science Bibliography. Additionally, the query was performed on Institute of Electrical and Electronics Engineers xplore (IEEExplore), ScienceDirect, SpringerLink, Wiley Online Library, and Association for Computing Machinery (ACM).

Table III lists the five resulting papers all of which were part of the literature review. Papers 1 [21], 3 [22], and 5 [24] are published papers that involve global software engineering and Agile methods. Papers 2 [19] and 4 [23] perform review of all globally DSE projects that include various development methodologies including Agile methods.

In paper 1, Jalali and Wohlin [21] 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

Table II. Search terms used to find distributed Agile software engineering (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)

DOI: 10.1002/smr

Table III. Summary of literature review.

ID	Reference	Title	Published	Year
1	[21]	Agile practices in global software engineering – a systematic map	IEEE – International Conference on Global Software Engineering (ICGSE)	2010
2	[19]	Empirical evidence in global software engineering –a systematic review	Association for Computing Machinery (ACM) – empirical software engineering	2009
3	[22]	Global software engineering and Agile practices – a systematic review	Wiley Journal of Software Maintenance and Evolution –researsh and practice	2011
4	[23]	Problems and solutions in distributed software development – a systematic review	SpringerLink – Software engineering approaches for offshore and outsourced development	2009
5	[24]	Using Scrum in global software development – a systematic literature review	IEEE – Fourth IEEE International Conference on Global Software Engineering	2009

IEEE, Institute of Electrical and Electronics Engineers.

of this study. A common result of papers reviewed by Jalali and Wohlin [21] 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 test-driven development (TDD).

In paper 3 [22], 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 global software engineering (GSE).

Paper 5 [24] focused on understanding the challenges faced when Scrum is used in DSE. It also helps to 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 [19] 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 performed for a few years and outsourcing had been practiced for up to 20 years, there was no single recipe for success. The outsourcing field is still relatively new, and, as such, there is a lack of methods, techniques, and tools in an industrial context [15]. 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 [19], paper 4 [23] 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

DOI: 10.1002/smr

CMM levels or could follow different frameworks (ITIL versus 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 because 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 [21, 22]. 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, that is, 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 [18], 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.

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 IX of [25].

4.2. Data sources and search strategy

The process of identifying relevant papers in the field of DASE 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 [19, 21–24].

DOI: 10.1002/smr

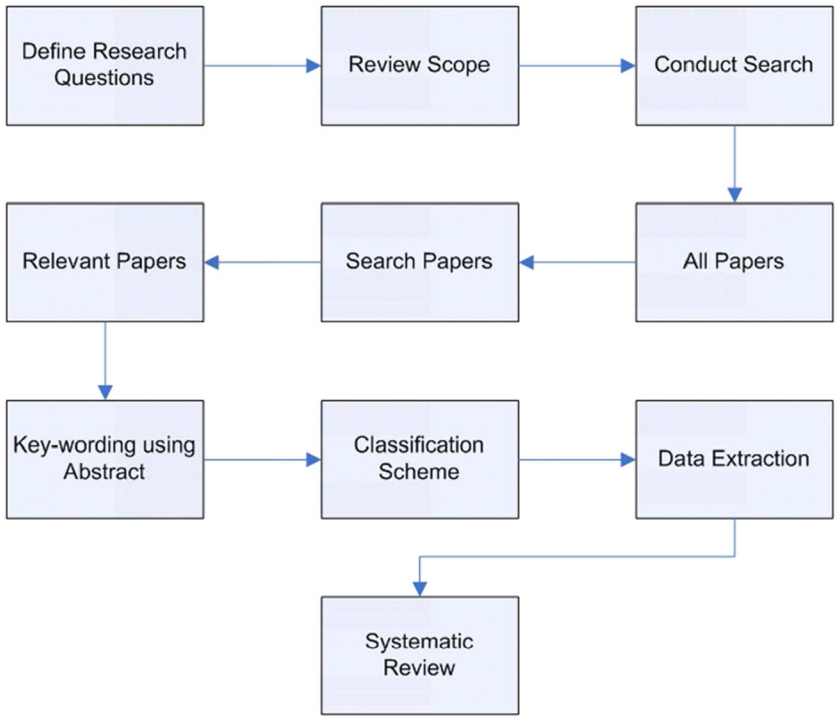

Figure 1. Systematic literature review phases.

Search criteria were set up based on the search query used in previous DASE systematic reviews [19, 21–24]. 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 Dorairaj *et al*. [26]. The search query below was used in the data sources listed in Table IV. The query was modified for each of the data sources previously 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 IV.

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 DASE 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 V and VI detail the inclusion and exclusion criteria, respectively.

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 [18] were used to appraise the attributes of the research design and reporting of the selected studies.

Table IV. Search terms used to find distributed Agile software engineering (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*)

DOI: 10.1002/smr

Table V. 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	Becuse 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	Because 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	Because 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 distributed Agile software engineering (DASE).

Table VI. 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 that 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.

The abstract was reviewed to understand if the problem participants, method of research, findings, and conclusion of the study were mentioned. Based on [18], 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 [18], the interpretation of analysis was reviewed to ensure all variables were accounted in the results. Because 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 [18, 27].

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

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

DOI: 10.1002/smr

Table VII. 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?

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 uniform resource locator, document object identifier, and authors.

The purpose of collecting the aforementioned information was to provide analysis of the metadata 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 uniform resource locator or document object identifier, 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 were also extracted. Data extracted were 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 [18]. Study environment is limited to academic and industrial. Additionally, the goal of the paper and the subjects was 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.

DOI: 10.1002/smr

Property 5 was used to extract data to answer RQ 3.1 and RQ 3.2. Based on the aforementioned variables, it was useful to see if the project was considered successful. There might be room for bias because 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.

In the succeeding texts, we provide further details on traceability between research questions and properties:

RQ 1.1 aims to bring forth the systems development life cycle (SDLC) phase involved in DASE. This was answered by understanding the context of the study (properties 1.1 and 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 and 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, and 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 work-arounds or mitigation strategies that had been put in place. Details were captured by reviewing work-arounds and mitigation plans documented in the papers (property 4.2). Work-arounds 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 on the information extracted as part of properties 4.3 and 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 properties 3.2 and 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 [19].

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.

DOI: 10.1002/smr

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, five on Wiley Online, and 43 on ACM portal. Implementing the inclusion and exclusion criteria, as outlined in Tables V and VI, 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 VII. 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 are shown in Figure 2.

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 V and VI. 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 VII. 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 [18, 28, 27]. However, it was [18] that was the primary source of guidance.

5.2.2. Validation of publication and primary study selection. As mentioned by Kitchenham *et al.* [18], 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,

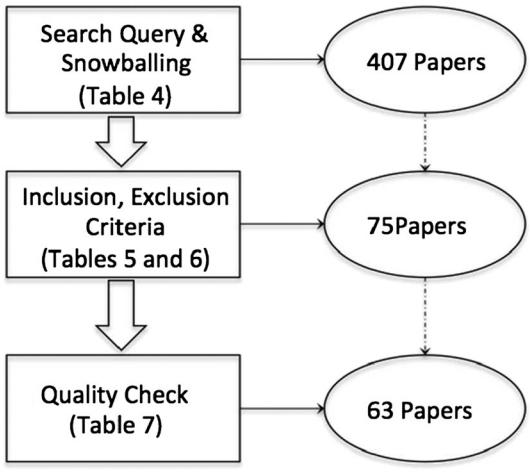

Figure 2. Inclusion process and results.

DOI: 10.1002/smr

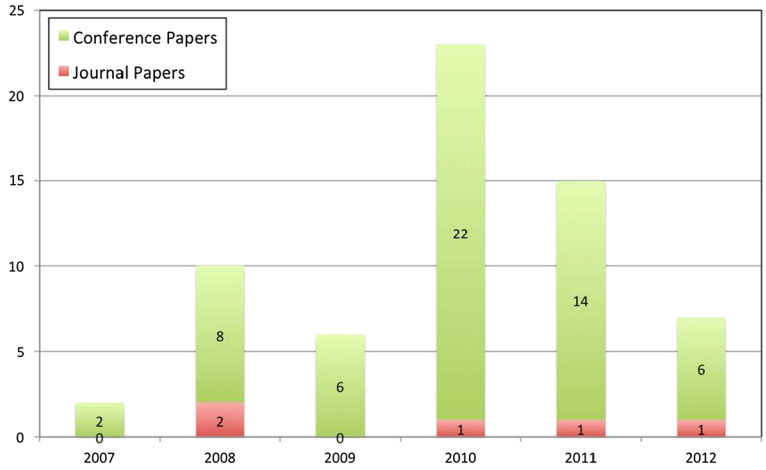

Figure 3. Distribution of selected studies by type.

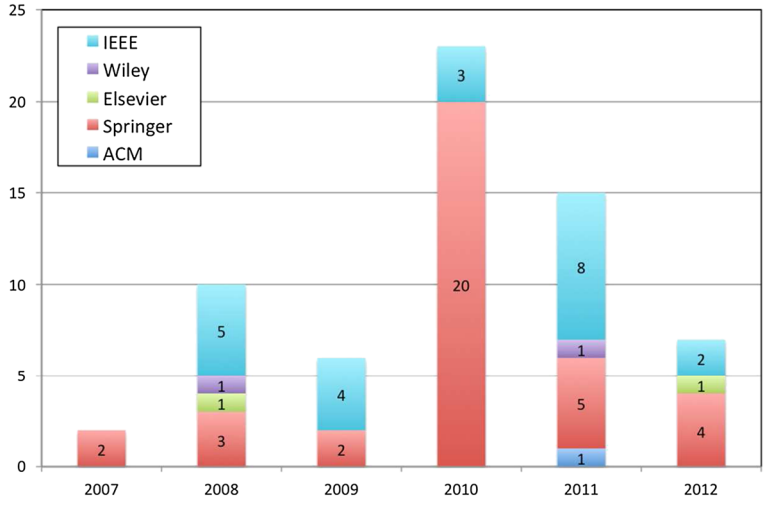

Figure 4. Distribution of selected studies by digital library. IEEE, Institute of Electrical and Electronics Engineers; ACM, Association for Computing Machinery.

there are several strategies that could be put in place including scanning gray literature, scanning conference proceedings, and contacting experts and researchers working in the area. We should point out that gray literature such as organization white papers and lessons learned was reviewed manually to address bias in paper selection.

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 aforementioned libraries. 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 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.

DOI: 10.1002/smr

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 [29]. 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. Because the authors' experience is not considered to be a credible source of information, reviewing published papers helped define the enumerations.

The data extraction properties 4 and 5 were sourced by literature review performed by [19, 21–24]. Additionally, early review of papers helped define preliminary guidelines.

Data classification proved to be without certainty because 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, Kitchenham *et al*. [18] recommend contacting the author of a questionable study to assist in resolving uncertainties and provide clarity to unknowns. However, Biolchini *et al*. [30] provide 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 Kitchenham *et al*. [18] 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 DASE 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 IV, 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 Motivation and Research Questions section.

6.1. What are the conditions under which organizations choose to adopt distributed Agile software engineering?

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.

DOI: 10.1002/smr

6.1.1. What phase of the systems development life cycle 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 VIII, '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 VIII are those that explicitly mention that they have only contributed to one of the listed SDLC phases in Table VIII, therefore, Table VIII 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 DSD.

6.1.2. What is the typical human resource distribution model (RQ1.2)?. This question provides an idea of the number of teams that was 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 IX summarizes the number of teams used in DASE projects. More than 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 were to save cost and access talent [31, 32]. This is primarily performed by engaging an additional team [32]. It was noted in one case where 12 teams were engaged on a project [33]. The scope of that project involved creating complex software for a multinational software development firm.

Table X summarizes the time difference between teams. In terms of time difference between teams, two projects [13, 34] were implemented using distributed human resources – both in the same time zone. These projects were academic and were implemented using students. Three projects [5, 7], and [35] had a time difference of 1 to 3 h. These projects were implemented within Europe. The 25% of the projects had teams that were (at peak) 3 to 5 h 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 VIII. 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%

QA, quality assurance.

Table IX. 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%

DOI: 10.1002/smr

Table X. Time zone difference.

Time difference	Total projects	%
Not mentioned	25	37%
3 to 4:59 PM	17	25%
9 to 10:59 PM	9	14%
5 to 6:59 PM	6	9%
7 to 8:59 PM	5	8%
1 to 2:59 PM	3	4%
0 h	2	3%

The distribution of the number of teams and the peak time zone between the teams is two important factors that can show how deeply DASE has been deployed in practice. As shown in Tables IX and X, from among the projects that reported these data, the majority of the projects were inclined towards a smaller number of teams, for example, two or three teams and also the peak time zone difference were 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 have already been reported by Jalali and Wohlin [22] (Figure 4).

Table XI provides us with an idea of why organizations choose to engage in DASE. The 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. The 12% had implemented DASE for experimentation. These were primarily academic projects. The 5% had engaged in Agile using distributed human resources because they had heard of the benefits of Agile or distributed human resources. The 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 [36–39]. Organizations chose to perform feasibility first to ensure that they have the capability to handle Agile using distributed human resources. In the case of [36, 38], organizations learned of how to streamline their processes on future DASE projects.

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 regard to distributed and Agile development. Table XII

Table XI. Reasons for engaging in distributed Agile software engineering (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%

Table XII. Experience in distributed Agile software engineering (DASE).

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

DOI: 10.1002/smr

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 XIII 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.

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 XIV summarizes the experience of human resources with Agile methods.

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 XV summarizes the collocation strategies implemented on DASE projects.

Table XIII. 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%

Table XIV. Experiences in Agile.

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

Table XV. 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%

DOI: 10.1002/smr

6.2. What are the biggest threats when adopting distributed Agile software engineering (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 performed by [40–47].

6.2.1. What are some of the biggest risks in distributed Agile software engineering (RQ 2.1)?. This question aims at documenting risks that projects have faced and issues that have risen during the course of the project.

6.2.1.1. 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 [48, 49]. In cases where time difference was more than 5 h, 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 [50, 38, 51]. Telephone was considered a good choice but was found effective only if both parties knew each other already. Because 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 [52]. This led to a *single-point-of-communication* way of information exchange: the project leaders [52].

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 [48], [32], [53], [54], and [55]. In some cases, the network connections between offices were not fast enough for video conferencing, or offices did not have video conferencing capabilities [56].

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 [44], [57], and [46].

Because 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 was lost through this communication medium [58], [32], and [46]. 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 [36], [40], [59], [60], [45], [61], [62], and [55].

When it comes to reviewing demo and sprints with all teams involved, software was not always used to demonstrate sprint reviews. This is because not all organizations had an *infrastructure setup*. 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 [38] and [53].

6.2.1.2. 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 was listed as some challenges.

One of the concerns under collaboration was that Scrum masters did not facilitate Scrum reviews [38]. 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 [38].

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 [63], [41], [64], [65], and [45]. Additionally, clients that did not have previous experience with Agile

DOI: 10.1002/smr

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 [45]. Teams that worked with a remote Scrum master and/or product owner were impacted on days when there was misunderstanding in scope [66].

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 [67] and [47]. 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 [35].

Lack of team structure and roles and responsibility is another challenge in DASE [63], [47], and [48]. This happens, more frequently, when team members lack experience. Agile practices state that every team member must collaborate as a generalist in project tasks [63]. 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 [47] and [53]. The detached nature of the customer and its representatives manifested itself especially when requirements that the remote team was accountable for were discussed [68].

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 [69], [26], and [62]. 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 carried out with distributed human resources. This is because dependencies required collaboration between team members who are working towards implementing the same stories [41], [64], [6], [11], [70], and [71]. Additionally, some work cannot be distributed because of 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 [43], [72], and [32]. This in turn 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 [33] and [55]. 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.

6.2.1.3. 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 on full documentation of requirements or product and sprint backlog [32]. As such, teams were at times *unclear on the requirements* that needed to be implemented. Requirements were gathered on exhaustive meetings (8 h duration) and documented in minutes [63] and [13]. Index cards were posted on walls at the office where requirement reviews were held and not always replicated manually at the other sites [39] and [66]. 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 [73]. 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

DOI: 10.1002/smr

industry standards. Such design could, at times, have dependencies between components [40], [33], and [53]. This leads to increased dependencies among products and components – with componentization being weak and code reuse being highly valued [33].

When dealing with very sensitive customer data, it was difficult for teams to pass data for testing or defect analysis [74]. 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 [74] and [70]. 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 [38], [52], [34], and [31]. This caused confusion within teams as vendor organizations were not used to the processes enforced by customers.

6.2.1.4. 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 varies 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 [46]. 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 [54]. 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 [37].

Language and practice are another area where cultural challenges apply. While communicating, it is noted that some cultures speak loud and direct, while some cultures are careful and cautions [40] and [59]. 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 [26].

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 [42]. This creates a false sense of hierarchy that some human resources are used to have – similar to having work assigned.

In regard 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 [33].

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

6.2.2. How are risks, limitations, and mitigation strategies in distributed Agile software engineering 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.

6.2.2.1. 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 [48], [58], [60], [7], [35], [71], and [61]. During Scrum meetings, teams could use video conferencing and utilize screen sharing when possible [48], [54], and [75]. 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.

DOI: 10.1002/smr

Table XVI. 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%

QA, quality assurance.

DOI: 10.1002/smr

As part of collaboration, team members were encouraged to use email to communicate when work hours did not overlap [47] and [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 [56]. 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 [60]. 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 e-mails should be used when asynchronous communication channel makes sense [73], [66], and [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 [47], [66], and [61]. Wiki-based Agile planning tools can also be utilized to publish, manage, integrate, and distribute agile planning information [66]. 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 [68].

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 [58], [50], [13], [53], [62], and [54]. An example of communication activities is an XP project that is undergoing the planning game or daily stand-up meetings [8]. The goal of a planning game is to get prioritized requirements from the customer [13], [65], and [8]. 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 [37], [59], [42], [60], [13], [73], [45], [35], and [76]. 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*. Because this is not feasible when working with distributed resources, it was recommended to have teams meet at times during the project [59], [67], [11], [7], and [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.

6.2.2.2. 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 [58] and [65]. 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

DOI: 10.1002/smr

scope correctly [59], [39], [57], [45], and [68]. 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 [31]. Papers suggest that a rigorous recruitment process should pay attention to both technical competence and cultural fit [31].

6.2.2.3. 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-h time zone difference, overlapping work timings will be very expensive and not feasible [50], [40], [67], [44], [57], [46], and [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 h, 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 [67], [69], [66], and [53].

Monitoring work is another way of dealing with issues. Iteration/sprint review sessions and daily scrum can be used to monitor work progress [50] and [52]. Additionally, senior resources can assist with code reviews, test case execution, and so on 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 [60] and [72]. 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 [37] and [56]. 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 [37], [59], [77], [45], [31], [7], and [61]. 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 [59], [31], and [7].

While most development work can be distributed, there is project work that is not easily carried out 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 carried out because of remote access restrictions, testing using data that cannot be shared with unauthorized team members, or work that is considered complex [74], [33], and [76].

Tools, if used correctly, can ease project coordination. This is especially true in DASE [40], [59], and [56]. 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 [40], [59] ,and [56].

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 [76].

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

DOI: 10.1002/smr

made available to the customer [6] and [71]. It was also noted that overall quality, interaction, cooperation, and experience improve 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 [74], [77], [6], [7], [56], and [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. Due to this aspect, both geographical and temporal distances become less of an issue beause developers can take certain decisions without having to confer with management, which could be located in another part of the world [78], [74], [79], and [8]. Additionally, the idea of applying semi-self-organized teams serves as a motivational goal [74] and [79].

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 because communication is considered a challenge when working with distributed resources [63], [52], [44], [32], and [8]. Any documentation that would decrease communication issues in an Agile method should be part of the process.

Having teams setup such that one (collocated) team works on a story together was considered best practice [11]. 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 is different and often initially uncomfortable in corporate contexts [80] and [72]. 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 [72].

Table XVII summarizes work-around and mitigation plans documented in published literature. Work-around 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 XVI and XVII, 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 XII. 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.

DOI: 10.1002/smr

Table XVII. Work-arounds and mitigation plans.

Risk/mitigation	Communication	Collaboration	Coordination	Cultural differences	Times reported	Percentage
Infrastructure (telephone, video conferencing, webcam, e-mail, 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%

DOI: 10.1002/smr

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 distributed Agile software engineering (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 XVIII summarizes Agile models used in projects. Based on the 63 papers reviewed, 40% of the project had used Scrum. The 14% had tailored and created a custom methodology called Scrum and XP. The 14% projects had used XP. One thing to note is that ScrumUP, a custom methodology, was created by one organization [81].

In terms of success, it is difficult to provide meaningful analysis as none of the projects was reported to have failed. Table XIX 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 was reported in Tables XV and XVII 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 two papers each

Table XVIII. 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%

Table XIX. 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

DOI: 10.1002/smr

and therefore the evidence is not conclusive, the other four methods mentioned in the literature typically had the same frequency in reporting the set of risks and mitigation strategies as shown in Tables XVI and XVII. 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.

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 XIX, 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 aforementioned 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 XX show the most common data collection method used in DASE research are observations represented by 33 papers (52%) followed by interviews in 19 papers (30%). In three papers (6%), a combination of observation and interviews was used, while in three papers (5%), a combination of research and reviewing documentation was used. When it came to reviewing documentation, researchers reviewed sources such as e-mails, communication logs, and wiki to capture data. Experience and surveys were used in one paper each, at 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 because research

Table XX. 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%

DOI: 10.1002/smr

within immature disciplines tends to be more exploratory in nature than research in mature fields that focus more on testing hypothesis, methods, or tools [19].

A significant number of works were conducted in the industry represented by 51 papers (81%), while eight papers (13%) were conducted in an academic setting. Equally, the number of studies used employees as subjects represented by 51 papers (81%), while eight 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. The 21% of the papers evaluated a practice, 9% evaluated a tool, 3% evaluated a method, and 2% evaluated a framework.

Table XXI 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.

The distribution in Table XXI 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 eight papers (13%) being evaluated in an industry setting.

Based on this distribution in Table XXI, it is fair to conclude that this review found a high degree of realism in the reviewed papers. A majority of challenges and work-arounds 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. Because 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.

Table XXI. 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%

DOI: 10.1002/smr

7. RECOMMENDATIONS

Based on our observations of the reviewed publications in the area of DASE, we find that although wealth of strong evidence is already provided for DASE, there are still a number of issues that need 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 performed 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. Because 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 previously, 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*: The 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 a situation when the DASE adoption strategy is strong but the project fails as a result of 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*: The 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 work-arounds 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.

DOI: 10.1002/smr

- *Collocation of teams*: Collocation helps build trust and improves working relationship as the project progresses. Collocation was deemed as the second best work-around 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 aforementioned 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 the following:

   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 carried out 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 are 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. Because 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 h 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 performed 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

DOI: 10.1002/smr

methodologies [6]. Ways in which these methods were tailored were 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 previously, 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 eight times (3%) as being a work-around to collaboration-related challenges. However, only one paper had performed an experiment to assess success in DASE [33]. 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. Because very little was captured in regard 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

Table AI. Primary studies and the results summary.

Paper number in Reference list	Study	Phase	Research Method	Agile method	Quality
[48]	How Agile practices have been tailored for adoption to distributed development and supporting GSD practices employed?	All phases	Interviews	Scrum	5.0
[58]	Understand how DASE works on large projects	All phases	Interviews	Scrum	12.0
[50]	Do Scrum practices provide any advantages over traditional software engineering methods when used in globally distributed projects?	All phases	Interviews	Scrum	12.0
[36]	Isolate and focus on the role of auditors, tools, and testers on distributed projects using Agile and Scrum	QA	Observation	Scrum	5.0
[37]	Understand how DASE performs when executed in overall systems development life cycle (SDLC)	All phases	Observation and Interviews	Scrum	6.5
[81]	To understand if a new custom methodology, ScrumUP, is an analysis for feasibility in GSD	All phases	Observation	scrumUP	8.0
[40]	To understand if success can be constantly achieved in DASE	All phases	Observation	Scrum and XP	7.0
[38]	How well do Agile and Scrum practices support distributed teams using tools to implement?	All phases	Observation	Scrum	10.5
[63]	Describes the experience of two large globally distributed companies implementing Scrum	All phases	Observation	Scrum	6.5
[41]	How does DASE work on a long-term project?	All phases	Interviews	Scrum	7.5

(Continues)

DOI: 10.1002/smr

Table AI. (Continued)

Paper number in Reference list	Study	Phase	Research Method	Agile method	Quality
[59]	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
[78]	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
[42]	What are the best practices when adopting DASE?	All phases	Observation	Scrum	4.0
[5]	To understand if the medium and method of communication effects success in DASE	All phases	Observation and interviews	Agile, specifics not mentioned	5.0
[43]	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
[64]	Captures the experience of a vendor house in handling distributed Agile projects.and 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
[60]	Describes experience of key resources in a large DASE project and explains how a team was able to make changes that allowed the continuous conversations to take place	Development	Observation	XP	8.0
[13]	Paper proposes flow mapping, a systematic approach for planning and managing information flows in distributed projects.	All phases	Observation	XP	13.5
[67]	Describes how the Agile coaches can help team members adopt Agile practices	All phases	Observation	Scrum	7.5
[74]	Using experience from two globally distributed outsourcing partners, this paper analyzes a comprehensive test automation strategy for their Agile teams that effectively leveraged both in-house and outsourced activities.	QA	Observation	Scrum	9.0
[80]	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
[77]	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

(Continues)

DOI: 10.1002/smr

Table AI. (Continued)

Paper number in Reference list	Study	Phase	Research Method	Agile method	Quality
[6]	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
[52]	'Papers show best practices when dealing with the lack of communication in a distributed Scrum team.'	All phases	Observation	Scrum	7.0
[39]	Paper reviews a distributed card-based planning tool (because physical cards are not available in DASE).	Requirement	Observation	Agile, specifics not mentioned	12.5
[82]	Best practices in collaboration that can improve success in DASE	All phases	Observation	Scrum and XP	10.5
[44]	Explains how trust can determine 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
[65]	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
[73]	This paper attempts to understand communication in DASE.	All phases	Observation	Agile, specifics not mentioned	10.0
[34]	Proposes a process that adds a level of governance to improve success in DASE	All Phases	Observation	XP	10.0
[11]	Conducts a study to see if Agile can be adopted on teams that do not share programming responsibility	All phases	'Survey'	–	11.0
[72]	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
[57]	Documents best practices to improve coordination in distributed Agile projects	All phases	Observation	XP	10.0
[45]	Conducts a detailed study of a software development organization following Scrum for developing software products using distributed resources	All phases	Observation	Scrum	7.0
[69]	Documents best practices in DASE	All phases	N/A	Scrum	3.5
[31]	Documents potential area of issues when engaging in Agile using distributed resources	All phases	Interviews	Agile, specifics not mentioned	9.5
[32]	Documents strategies that project managers can use when working on Agile projects using distributed resources	All phases	Interviews	Agile, specifics not mentioned	11.0
[83]	Shows how planning can be improved in DASE	Planning	Interviews	Scrum	9.5

(Continues)

DOI: 10.1002/smr

Table AI. (Continued)

Paper number in Reference list	Study	Phase	Research Method	Agile method	Quality
[46]	Effects of culture, competence, and knowledge asymmetry in DASE	All phases	Observation and interviews	Scrum and XP	6.0
[70]	Constructs a preliminary conceptual model for exploring three proposed dimensions necessary for successful configuration of global Agile teams, structure, agility, and virtualness	All phases	Observation and interviews	Scrum and XP	7.0
[79]	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
[33]	Explains the issues faced when going from Waterfall to Agile on large distributed projects	All phases	Observation	Lean development and Scrum	10.0
[84]	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
[7]	Documents lessons learned from projects that have implemented Agile using distributed resources	All phases	Interviews	Agile, specifics not mentioned	9.0
[47]	Highlighting challenges and success during My Yahoo development – DASE	All phases	Observation	Scrum	9.5
[56]	Shows the importance of tools when developing a product using pair programming using distributed resources	Development	Observation	Pair programming	6.0
[26]	Documents key concerns in DASE	All phases	Interviews	Scrum	9.5
[68]	Provides guidelines that organizations can follow in DASE	All phases	Interviews	Scrum	10.0
[8]	Paper outlines 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
[85]	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
[66]	Evaluates tools that could be used for planning in DASE	Planning	Observation	Agile, specifics not mentioned	5.0
[35]	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.	Al phases	Observation	Scrum	11.5
[86]	Paper investigates how Agile teams can be distributed by adding a 'remote partner' and still maintain Agile advantages.	All phases	Observation	XP	6.5

(Continues)

DOI: 10.1002/smr

Table AI. (Continued)

Paper number in Reference list	Study	Phase	Research Method	Agile method	Quality
[49]	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
[71]	Experience of a global multinational company on transitioning from distributed and traditional to distributed and Agile	All phases	Interviews	Scrum	8.0
[61]	How to work with vendors/ subcontractors in DASE	All phases	Interviews	Scrum	10.0
[76]	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
[62]	How to work with a vendor in DASE where the vendor has a higher capability maturity model integration(CMMI) level?	All phases	Interviews	Scrum	7.0
[51]	Paper describes how Scrum practices could be successfully applied in a distributed setting.	All phases	Interviews	Scrum and XP	12.0
[54]	Explains how a typical DASE project can face particular control challenges related to balancing fixed versus evolving quality requirements and people versus process-based collaboration.	All phases	Interviews	Scrum and XP	11.0
[55]	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
[75]	Paper analyzes 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

GSD, global software development; QA, quality assurance; DASE, distributed Agile software engineering; GSE, global software engineering; GDSD, globally-distributed software development.

APPENDIX: B

Table BI. 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%

(Continues)

DOI: 10.1002/smr

Table BI. (Contiuned)

Criteria	Yes	No	Somewhat
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

Table CI. Data extraction properties.

Number	Property	Values	Paper selection	RQ mappings
1	Is the context of the study described? 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?	2.1 – (Business practice, experiment, heard of Agile, not mentioned, to simulate)	Background	RQ 1.1, RQ 1.2, RQ 1.3
	2.2 What phase of the project lifecycle has utilized distributed human resources?	2.2 – (All phases, design, development, planning, QA and testing, requirements)		
	2.3 What is the human resource distribution model?	2.3 – Number of teams and how far are they located		
	2.4 Do human resources have experience in distributed Agile software engineering?	2.4 – (Yes, no, some resources, not mentioned)		
	2.5 Do human resources have experience in distributed teams?	2.5 – (Yes, no, some resources, not mentioned)		
	2.6 Do human resources have experience in Agile?	2.6 – (Yes, no, some resources, not mentioned)		
	2.7 Have human resources been collocated at some point during the project lifecycle?	2.7 – (Seed visits, maintenance visits, seed and maintenance visits, no, not mentioned)		
3	3.1 What type of research method has been used in this study?	3.1 – (Experience, interview, not mentioned, observation, observation and interviews, research and documentation, survey)	Research method	RQ 4.1, RQ 4.2
	3.2 In which environment has this study taken place?			
	3.3 Who are the subjects of this research?	3.2 – (Industry, academic, not mentioned)		
	3.4 Does the paper evaluate a practice, method, tool, or framework?	3.3 – (Employees, students, not mentioned)		
		3.4 – (Practice, method, tool, framework, capture risks/mitigation)		
4	4.1 Are risks and issues documented?	4.1 – (A list of challenges, risks, and issues documented)	Results	RQ 2.1, RQ 2.2, RQ 3.1
	4.2 Are work-around or mitigation plans listed?	4.2 – (A list of solutions, work-around, and mitigation plan)		
	4.3 What type of Agile model was used?	4.3 – (Agile, specifics not mentioned, Scrum, XP, Scrum and XP, Lean development and Scrum, pair programming, ScrumUp)		

(Continues)

DOI: 10.1002/smr

Table CI. (Continued)

Number	Property	Values	Paper selection	RQ mappings
	4.4 Was there a specific reason to use an Agile model?	4.4 – (Business practice, experience, experiment, knowledge within the team, research, not mentioned)		
5	5.1 Was the project considered a success? 5.2 Will the organization engage in DASE?	5.1 – (Yes, no, somewhat, not mentioned) 5.2 – (Yes, no, not mentioned)	Conclusion	RQ 3.1, RQ 3.2

DASE, distributed Agile software engineering; QA, quality assurance.

REFERENCES

1. Therrien E. 'Overcoming the challenges of building a distributed Agile organization', AGILE '08. Conference, 368–372, 2008. DOI: 10.1109/Agile.2008.9
2. Smite D. et al. 'Empirical evidence of global software engineering: a systematic review' Empirical software engineering, 2010. DOI: 10.1007/s10664-009-9123-y
3. Young C, Terashima H. 'How Did We adapt Agile processes to our distributed development?' Agile 2008, pp. 304–309, 2008. DOI: 10.1109/Agile.2008.7
4. Martin RC. *Agile Software Development: Principles, Patterns, and Practices*. : Prentice Hall PTR, 2003.
5. 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. DOI: 10.1109/ICGSE-W.2011.15
6. 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. DOI: 10.1145/1987875.1987894
7. Paasivaara M, Lassenius C. 'Using Scrum practices in GSD projects' *Agility across Time and Space*, 2010, (4), 259–278, 2010. DOI: 10.1007/978-3-642-12442-6_17
8. Maruping LM. 'Implementing extreme programming in distributed software project teams: strategies and challenges' *Agility across Time and Space, 2010, Part 2*, 2010; 11–30. DOI: 10.1007/978-3-642-12442-6_2
9. Zopf S. 'Success factors for globally distributed projects' *Software Process: Improvement and Practice*, 355–359, 2009. DOI: 10.1002/spip.423
10. Munch J. et al. 'Understanding lacking trust in global software teams: a multi-case study' *Product-Focused Software Process Improvement*, 20–34, 2007. DOI: 10.1007/978-3-540-73460-4_6
11. Korhonen K. 'Adopting Agile practices in teams with no direct programming responsibility: a case study' Lecture Notes in Computer Science, 2011, 6759, Product-Focused Software Process Improvement, 30–43, 2011. DOI: 10.1007/978-3-642-21843-9_5
12. Paasivaara M, Lassenius C. 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.
13. 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. DOI: 10.1109/ICGSE.2011.9
14. Estler H-C, Nordio M, Furia CA, Meyer B, Schneider J. Agile vs. structured distributed software development: a case study, global software engineering (ICGSE), 2012 IEEE Seventh International Conference on, 2012 **11**(20): 27–30.
15. Fitzgerald B. et al. 'Customising Agile methods to software practices at Intel Shannon' *European Journal of Information Systems* **15**, 200–213. 2006. DOI: 10.1057/palgrave.ejis.3000605
16. West D, Grant T. 'Agile development: mainstream adoption has changed agility', Forrester, 2010, (Available from: http://pmshow2012.programmedevelopment.com/public/uploads/files/forrester_agile_development_mainstream_adoption_has_changed_agility.pdf [May 7, 2015]).
17. Dedrick J, Kraemer KL, Dunkle D. 'Offshoring of software development: patters and recession effects', Recent Work, Personal Computing Industry Center, University of California – Irvine, 2010, (Available from: http://escholarship.org/uc/item/22r9q7wp [May 13, 2015]).
18. 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* 2002; **28**(8):721–734.
19. Smite D, Wohlin C, Gorschek T, Feldt R. 'Empirical evidence in global software engineering: a systematic review', *Empirical Software Engineering*, **15**(1), 2010, 91–118. DOI: 10.1007/s10664-009-9123-y
20. Jiménez M, Piattini M, Vizcaíno A. Challenges and improvements in distributed software development: a systematic review. *Advance Software Engineering* 2009; Article 3 (January 2009):16. DOI: 12009/2009/710971

21. Jalali S, Wohlin C.'Agile practices in global software engineering – a systematic map', 2010 5th IEEE International Conference on Global Software Engineering (ICGSE), 45–54, 2010. DOI: 10.1109/ICGSE.2010.14
22. 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* **24**(6): 643–659, 2012. DOI: 10.1002/smr.561
23. 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, **16**, 107–125. DOI: 10.1007/978-3-642-01856-5_8
24. Hossain E, Babar MA, Paik HY. 'Using Scrum in global software development: a systematic literature review', ICGSE 2009. Fourth IEEE International Conference on Global Software Engineering, 2009, 175–184. DOI: 10.1109/ICGSE.2009.25
25. Kitchenham BA. 'Procedures for performing systematic reviews', tech. report SE0401, Dept. of Computer Science, Univ. of Keele, and tech. report 0400011 T.1, Empirical Software Eng., National Information and Communications Technology Australia, 30 Aug. 2004.
26. Dorairaj S, Noble J, Malik P. 'Understanding team dynamics in distributed Agile software development' Lecture Notes in Business Information Processing, 2012, 111, *Agile Processes in Software Engineering and Extreme Programming, Part 2*, 2012; 47–61. DOI: 10.1007/978-3-642-30350-0_4
27. Jedlitschka A, Ciolkowski M, Pfahl D. 'Reporting experiments in software engineering', in Shull F, Singer J, Sjøberg D (eds). '*Guide to Advanced Empirical Software Engineering*', 2008, Springer-Verlag: London, chapter 8.
28. Runeson P, H'ost M. 'Guidelines for conducting and reporting case study research in software engineering'. *Empirical Software Engineering*, 131–164, 2009.
29. Wright RW, Brand RA, Dunn W, Spindler KP. How to write a systematic review. *Clinical Orthopaedics and Related Research* 2007; **455**:23–29.
30. Biolchini J, Gomes P, Cruz A, Travassos G. *Systematic Review in Software Engineering*. Systems Engineering and Computer Science Department, UFRJ: Rio de Janeiro, Brazil, 2005.
31. Srinivasan J. 'Preparing your offshore organization for agility: experiences in India' *Agility across Time and Space 2010, Part 2*, 2010; 117–130. DOI: 10.1007/978-3-642-12442-6_8
32. Hossain E, Babar MA, Verner J. 'Towards a framework for using Agile approaches in global software development' *Lecture Notes in Business Information Processing*, 2009, **32**, 4, 126–140, 2009. DOI: 10.1007/978-3-642-02152-7_11
33. Schnitter J, Mackert O. 'Large-scale Agile software development at SAP AG' *Communications in Computer and Information Science, 2011*, **230**, 209–220, 2011. DOI: 10.1007/978-3-642-23391-3_15
34. Klein H, Knauss K, 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, 71–83, 2012. DOI: 10.1007/978-3-642-27213-4_6
35. 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. DOI: 10.1007/978-3-642-13054-0_39
36. 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. DOI: 10.1109/CSEET.2011.5876097"
37. Su SH, 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. DOI: 10.1109/ICCSE.2010.5593595"
38. 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. DOI: 10.1109/ICSEA.2010.66
39. 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. DOI: 10.1007/978-3-540-68255-4_6
40. 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. DOI: 10.1109/Agile.2008.92"
41. Paasivaara M, Lassenius C. 'Scaling Scrum in a large distributed project', 2011 International Symposium on Global Software Engineering, 2009, 2011. DOI: 10.1109/ESEM.2011.49
42. Hansen MT, Baggesen H. 'From CMMI and isolation to Scrum, Agile, Lean and collaboration' Agile Conference, 2009. AGILE '09., 2009. DOI: 10.1109/AGILE.2009.18
43. Alyahya S, Ivins WK, Gray WA. 'Co-ordination support for managing progress of distributed Agile projects' 2011 Sixth IEEE International Conference on Global Software Engineering, 2009, 2011. DOI: 10.1109/ICGSE-W.2011.24"
44. 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, 172–177, 2010. DOI: 10.1007/978-3-642-13054-0_14
45. Inayat I, Muhammad AN, Inayat Z. 'Facilitating an off-site customer in product-based Agile software development: an industrial case study' *Communications in Computer and Information Science*, 2012, 281, 210–221, 2012. DOI: 10.1007/978-3-642-28962-0_21

46. 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*, 245–258, 2010. DOI: 10.1007/978-3-642-15346-4_20
47. Lee S, Yong HS. 'Distributed Agile: project management in a global environment' *Empirical Software Engineering* 2010; **15**(2), 2010. DOI: 10.1007/s10664-009-9119-7
48. 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. DOI: 10.1109/ICGSE.2009.27
49. 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. DOI: 10.1007/978-3-642-20677-1_8
50. Bannerman PL, 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. DOI: 10.1109/HICSS.2012.512
51. Paasivaara M, Durasiewicz S, Lassenius C. 'Using Scrum in a globally distributed project: a case study' *Software Process: Improvement and Practice*, 2008: **13**(6), 527–544. DOI: 10.1002/spip.402
52. Bless M. 'Distributed meetings in distributed teams' Lecture Notes in Business Information Processing, 2010, **48**(3), 251–260, 2010. DOI: 10.1007/978-3-642-13054-0_27
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. DOI: 10.1007/978-3-642-24815-3_8
54. Persson JS, LarsMathiassen L, Aaen L. Agile distributed software development: enacting control through media and context. *Information Systems Journal* 2011. DOI:10.1111/j.1365-2575.2011.00390.x.
55. Drurya M, Conboy K, Power K. 'Obstacles to decision making in Agile software development teams' *Journal of Systems and Software* 2012; **85**(6): 1239–1254, 2012. DOI: 10.1016/j.jss.2012.01.058
56. Dajda J, Dobrowolski G. 'How to build support for distributed pair programming' Lecture Notes in Computer Science, 2007, **4536/2007**, 2007; 70–73. DOI: 10.1007/978-3-540-73101-6_10
57. Hole S, Moe NB. 'A case study of coordination in distributed Agile software development' *Communications in Computer and Information Science, 2008, 16, Software Process Improvement, Part 5* 2008; 189–200. DOI: 10.1007/978-3-540-85936-9_17
58. 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. DOI: 10.1109/ICGSE.2008.38
59. 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. DOI: 10.1109/HICSS.2009.225
60. Williams W, Stout M. 'Colossal, scattered, and chaotic (planning with a large distributed team)'Agile, 2008. AGILE '08. Conference, 2008. DOI: 10.1109/Agile.2008.25
61. Rudzki J, Hammouda I, Mikkola T, Mustonen K, Systä T. 'Considering subcontractors in distributed Scrum teams' *Agility across Time and Space, 2010, Part 3*, 2010; 235–255. DOI: 10.1007/978-3-642-12442-6_16
62. Kussmaul C. 'Onshore and offshore outsourcing with Agility: lessons learned' *Agility across Time and Space, 2010, Part 2*, 91–106, 2010. DOI: 10.1007/978-3-642-12442-6_6
63. 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. DOI: 10.1109/ICGSE.2008.34
64. Sureshchandra K, Shrinivasavadhani J. 'Adopting Agile in distributed development' ICGSE 2008. IEEE International Conference on Global Software Engineering, 2009, 2008. DOI: 10.1109/ICGSE.2008.25"
65. Kornstädt A, Sauer J. 'Mastering dual-shore development: the tools and materials approach adapted to Agile offshoring' Lecture Notes in Computer Science, 2007, 4716, *Software Engineering Approaches for Offshore and Outsourced Development*, 2007; 83–95. DOI: 10.1007/978-3-540-75542-5_7
66. Wang X, Maurer F, Morgan R, Oliveira J. 'Tools for supporting distributed Agile project planning' *Agility across Time and Space, 2010, Part 3*, 2010; 183–199. DOI: 10.1007/978-3-642-12442-6_13
67. Paasivaara M. 'Coaching global software development projects' 2011 6th IEEE International Conference on Global Software Engineering, 2009, 2011. DOI: 10.1109/ICGSE.2011.33"
68. Korkala M, Pikkarainen M, Conboy K. 'Combining Agile and traditional: customer communication in distributed environment' *Agility across Time and Space, 2010 Part 3*, 2010; 201–216. DOI: 10.1007/978-3-642-12442-6_14
69. Šmite D, Moe NB, Ågerfalk PJ. 'Fundamentals of Agile distributed software development' *Agility Across Time and Space*, 2010, 1, 3–7, 2010. DOI: 10.1007/978-3-642-12442-6_1
70. Sharp JH, Ryan SD. 'A preliminary conceptual model for exploring global Agile teams' *Lecture Notes in Business Information Processing, 2008*, **9**(6), 147–160, 2008. DOI: 10.1007/978-3-540-68255-4_15
71. Wildt D, Prikladnicki R. 'Transitioning from distributed and traditional to distributed and agile: An experience report' *Agility across Time and Space, 2010, Part 2*, 2010; 31–46. DOI: 10.1007/978-3-642-12442-6_3
72. Bosch J, Bosch-Sijtsema P. 'Coordination between global Agile teams: from process to architecture' *Agility Across Time and Space*, 2010, **3**, 217–233, 2010. DOI: 10.1007/978-3-642-12442-6_15

73. 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*, 2009; 161–167, DOI: 10.1007/978-3-642-01853-4_21
74. Little T, Elliott S, Hughes J, Simion F. 'Leveraging global talent for effective test agility' Agile Conference (AGILE), 2012, 2012. DOI: 10.1109/Agile.2012.25
75. Sharp H, Robinson H. 'Collaboration and co-ordination in mature eXtreme programming teams' *International Journal of Human-Computer Studies archive* 2008; **66**(7): 506–518, 2008. DOI: 10.1016/j.ijhcs.2007.10.004
76. Sadun C. 'Scrum and global delivery: pitfalls and lessons learned' *Agility across Time and Space, 2010, Part 2*, 2010; 71–89. DOI: 10.1007/978-3-642-12442-6_5
77. 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. DOI: 10.1109/ICGSE.2011.19"
78. 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. DOI: 10.1109/ICGSE.2009.22
79. Kumlander D. 'Motivating company personnel by applying the semi-self-organized teams principle' *Innovations in Computing Sciences and Software Engineering, 2010*, 245–248, **2010**, 10.1007/978-90-481-9112-3_41
80. Mudumba V, Lee O-K. 'A new perspective on GDSD risk management: Agile risk management' 2010 5th IEEE International Conference on Global Software Engineering, 2009, 2010. DOI: 10.1109/ICGSE.2010.33"
81. del Nuevo E, Piattini M, Pino FJ. 'Scrum-based methodology for distributed software development' global software engineering (ICGSE), 2011 6th IEEE International Conference on Global Software Engineering, 2009, 2011. DOI: 10.1109/ICGSE.2011.23
82. Sharp H, Giuffrida R, 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, 62–76, 2012. DOI: 10.1007/978-3-642-30350-0_5
83. 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*, 171–182, 2010. DOI: 10.1007/978-3-642-12442-6_12
84. 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*, **48**(2), 202–207, 2010. DOI: 10.1007/978-3-642-13054-0_19
85. Power K. 'Using silent grouping to size user stories' *Lecture Notes in Business Information Processing, 2011*, 2011; **77**(1), 60–72. DOI: 10.1007/978-3-642-20677-1_5
86. 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. DOI: 10.1007/978-3-642-16416-3_14