IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 10, 2021, accepted December 7, 2021, date of publication December 20, 2021,
date of current version January 25, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3136923

Distributed Agile Patterns-Using Agile Practices to Solve Offshore Development Issues

MARYAM KAUSAR, M. ISHTIAQ, AND SHARIQ HUSSAIN
Department of Software Engineering, Foundation University Islamabad, Islamabad 44000, Pakistan

Corresponding author: Maryam Kausar (maryam.kausar@fui.edu.pk)

This work was supported by FUI.

ABSTRACT As offshoring is becoming mainstream, companies are moving towards using agile methods. Offshoring has many advantages such as reduced development cost, proximity to market and round the clock development, it has created new challenges for the application of agile practices as the teams are now distributed. Companies are adapting and modifying agile practices in order to overcome these challenges. However, little effort has been put into identifying common practices, which are being repeatedly, used to solve frequent problems in offshore development. In this research, we have studied from literature over 200 cases and have interviewed professionals participating in distributed offshore teams. Based on the observations, we have designed a solution known as Distributed Agile Patterns, which will address the common issues in offshoring scenario. Fifteen distributed agile patterns have been identified and classified into four categories based on the type of problem they solve. A reflection workshop was conducted where professionals were invited to review the pattern catalogue and help us verify and validate the catalogue. Based on their feedback, the catalogue was finalised. The purpose of the catalogue is to serve as a guideline for practitioners to use to aid them in adoption of agile practices in offshoring.

INDEX TERMS Distributed agile patterns, offshore development issues.

I. INTRODUCTION

Offshoring work is not a new concept; many companies from cleaning services to consultations have been using it [1]. However, offshoring work internationally has increased with the help of technology as it can cut down communication cost with the help of online tools, allowing companies to work with remote teams at cheaper rates efficiently [2].

In the past mostly manufacturing companies were using offshoring, however, with cheaper and faster communication many companies from other sectors started to move towards offshoring [3]. Similarly, offshoring is changing how companies develop software. Reduction in cost is the main reason why companies opt for offshoring, which is mainly because of lower salaries in comparison with counties such as Europe, the US and Japan [4]. For example, companies would be 80% more economical to offshore work from UK to India [5]. Apart from the cost there are some other advantages to offshoring as well i.e. access to a pool of skilled people in a country like India that profits 90% from IT revenue as it produces 2 million university graduates per year [6]. When Kobayashi-Hillary went to Bangalore to start a company he states that the issue was now to separate an excellent resource from a good resource [6]. Another advantage is proximity to markets and customers [7], [8].

Due to the distribution of teams over different geographical locations arises the issue of trust, socio-cultural, communication and co-ordination, and knowledge transfer [8], [9]. These issues affect the working of distributed teams, which in turn affect the applicability of agile practices i.e. agile highly depends on face-to-face communication, which is not possible in distributed scenario.

Many teams have tried to experiment and adapt the agile practices to avoid the above-identified challenges however such efforts were individual and difficult to share with practitioners. This paper provides solutions to some of the challenges by identifying agile practices, which address these issues repeatedly in offshore scenario. We believe that by identifying and cataloguing such practices, practitioners can easily retrieve and apply them in similar situations. The solution is represented as the distributed agile patterns. We show that with the help of patterns, practitioners will be able to adopt agile easily for offshore development.

The associate editor coordinating the review of this manuscript and approving it for publication was Porfirio Tramontana.

Agile offers disciplined yet lightweight processes and since the formation it has changed how software is developed [10], [11]. The agile manifesto emphasises on customer satisfaction. It focus on keeping the customer satisfied and nowadays many companies are selecting agile to develop software [10]. It is observed that around 84% of the organisations are still maturing their agile practices and creating opportunities for growth [12].When companies opt to offshore software development, it is considered as a risk to cut down on cost of software development [13]. Such companies use agile methods to attain better planning and execution of the project [14]–[16].

Ghani *et al.* [17] stated, that in order to apply agile in offshore development companies need to overcome the 5 challenges that are communication, coordination, cooperation, collaboration and control. Similarly, Taylor *et al.* [18] states that in order to use agile processes in offshore development, projects have to undergo many variations in development practice such as communication becomes more formal, teams have to self manage between different offshore teams and cross-functional teams can be difficult to control [19]. Many systematic literature reviews have been conducted to study the use of agile practices in global software development [20]–[22]. It is clear that due to differences in offshore development, the use of agile in offshoring is not an easy process and many companies that opt for offshoring, customise agile processes in order to avoid offshore challenges.

An empirical study on the challenges of agile offshore development in which 334 practitioners took part, agreed that in order for agile to be successful for offshore projects, managers have to focus on choosing a team that is technically competent, effective in communication and coordination with the customer and continuously engages with the customer throughout the project and needs to manage a low staff turnover in teams [23]. Similarly a comparative analysis done on in-house and offshore development using agile methods to code and design software showed that companies are more comfortable to migrating to agile if it is applied to only one phase [24]. Efforts have been made to overcome the offshore challenges such as cultural differences. Šmite *et al.* [25] believes that by solving the challenges caused by cultural differences collaboration among the onshore and offshore team can improve.

The article is organized as follows, section II discuss the offshore software development challenges and how these challenges affect the agile principles. Section III presents the current work done on patterns and gives the overview of distributed agile patterns while section IV explains the research methodology used for the identification of the distributed agile patterns. Section V lists the results of the study, while section VI mentions the threats to the validity of the results and section VII, concludes the study.

II. CHALLENGES IN AGILE OFFSHORE SOFTWARE DEVELOPMENT

In this section the inherent challenges of offshore development have been identified and how the challenges affect agile practices. While conducting an extensive literature review on offshoring, many challenges were identified which we classified into four categories which are trust, socio-culture, communication & coordination, and knowledge transfer as most of them overlapped these four areas. These challenges occur when the teams are distributed over different time zones. The challenges have also been observed to affect agile adoption in offshore scenario. By classifying the challenges we can map their effect on different agile practices, this will help in constructing patterns that will solve them.

TABLE 1. Challenges identified in offshore software development.

No.	Challenge	Evidence
1	Trust issue	E1, E2, E3, E4, E5, E6, E7, E11, E13, E19, E20, E24, E28, E29, E30, E33, E38, E41, E42, E45, E46, E47, E49, E54, E56, E57, E62, E67, E68, E70
2	Socio-Cultural issue	E1, E2, E3, E4, E6, E7, E11, E12, E16, E18, E19, E24, E26, E27, E28, E31, E32, E35, E41, E42, E46, E47, E44, E48, E52, E62, E66, E67, E68, E71
3	Communication and Co-ordination issues	E1, E4, E6, E8, E9, E10, E16, E16, E18, E19, E21, E22, E23, E24, E25, E28, E36, E37, E39, E40, E41, E43, E46, E47, E48, E50, E52, E53, E55, E58, E59, E60, E61, E62, E63, E64, E65, E68, E69, E71
4	Knowledge Transfer issue	E1, E10, E14, E15, E22, E32, E34, E46, E47, E51, E53, E55, E60, E68

A. INHERENT CHALLENGES OF OFFSHORING

The Table 1 elaborates the results of a study that was conducted in which the inherent challenges of offshoring were identified. It shows how frequent the challenges occur in literature. The first column categorised the offshore challenges and in the second column provides the evidences from where the challenges were identified. The studies, which were selected for evidence, have been presented as additional material to this paper and it is available online here: https://cutt.ly/YWLRyYQ

As one of the main challenges in offshore development is **trust** [26]–[28] it is important to establish trust among firms for successful partnerships and alliances [29]. Trusted partnerships enable open exchange of information, which helps improve the response to any crises, which may occur during the executions of a project [30]. While realizing that trust is important in offshoring, it is difficult to establish, as it is difficult to trust unknown foreign partners, which are in different time zones [31]. Despite this difficulty, cooperating with foreign partners is very important and without trust, organisations tend to divert the responsibility rather than accepting ownership [32].

In offshore software development, **socio-cultural** issues arise due to differences in language, national traditions, values and norms, which adds another challenge to offshore development [19], [26], [33]. Socio-cultural distance is defined as the extent of how much each person understands one another [4]. In terms of offshore development, it becomes more complex as it consists of the culture of an organisation, the countries national traditions, local languages, government and political views and lastly even the work ethics of individuals [34]. A notable example of socio-cultural issue is the language, as English is not the first language in countries like India, Bangladesh, Pakistan and China, where a lot of

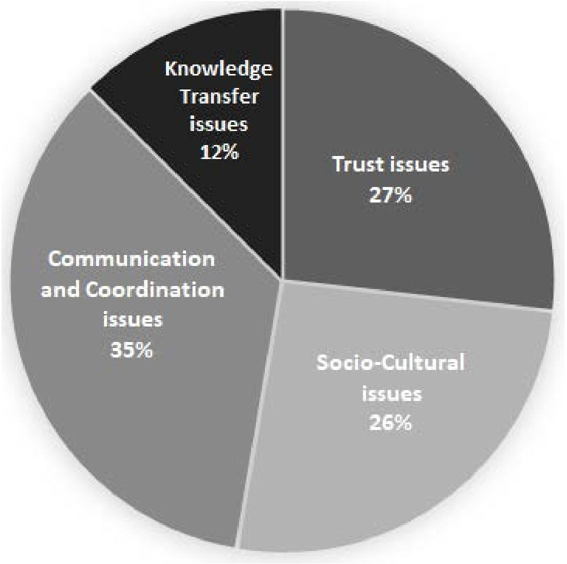


FIGURE 1. Overview of the offshore challenges.

work is offshored to; so extra effort is needed to communicate and coordinate between teams [35]. Any incorrect use of vocabulary can lead to miscommunication, which can result in adding hidden costs of correction [36].

At the earlier stages of software development a lot of communication is required but teams which are distributed geographically in different time zones face many **communication and coordination** issues as they get less real-time communication. It affects various aspects of the working partnership such as trust, relationships and efficiency of the team [19], [37]–[40]. A study done by Ebert [41] explained that approximately half of all the offshore projects fail due to the lack of sufficient communication, and trust between the team. A similar study done by Herbsleb *et al.* [42] showed that insufficient communication could drag the response of a problem, which causes projects to miss deadlines or fail.

Another issue is **knowledge transfer**, which introduces challenges in offshore software development [19], [43], [44]. It is difficult for managers to take advantage of offshoring if effective knowledge sharing processes are not set in place [8]. Knowledge is a key concept, and Davenport *et al.* [43] defines it as, any document or repository that an organization has, is considered as knowledge. In offshoring, knowledge transfer is a critical process as companies are moving their business culture and development processes to offshore locations and any mistake can cause a negative effect on the company. A study done by Radoff [45] states that the common problems between companies transferring knowledge are; problems in communication, different ways in which work is conducted and work attitudes, and how different is the decision making process. Based on the literature presented in Table 1 and the same has been summarised in Figure 1 it can be seen that communication and coordination issues along with trust, socio-cultural and knowledge transfer are the key factors for the success of offshore project.

TABLE 2. Affect of offshore challenges on agile practices.

No.	Offshore Challenge	Agile Practice	Affect of Challenge on agile practice
1	Trust	Collective ownership	Dispute over code ownership among the distributed team members.
		Sustainable pace	It's hard to maintain a sustainable pace of the offshore project.
2	Socio-cultural	Iterative and incremental development	Causes delays in the frequency of code delivery.
		Self-organising teams	Hard to form self-organising teams
3	Communication and Coordination	Sprint planning	Issues in project design as team is distributed
		Continuous integration	Too many versions of code
4	Knowledge Transfer	Product backlog	Failure in documentation of all the changes can cause the backlog to be incomplete.
		Sprint review	Tractability of progress is difficult as team is distributed.

B. THE EFFECT OF OFFSHORING ON AGILE ADOPTION

The four key challenges in offshore development were identified and how they affect the agile adoption process will be shown in this section. From the collection of systematic literature reviews conducted on agile offshore development [20]–[22], [46]–[48] and attempts to solve them [23]–[25]. Table 2 was designed. For example, if we consider the communication and coordination challenge, it affects agile practices such as sprint planning and continuous integration. It can be observed in Table 2 that how offshoring affects some agile practices:

As shown in Table 2, agile practices do not match well with offshore development, which is why practitioners have been trying to modify and adapt agile practices for offshore development. This research believes that with the help of patterns, practitioners will be able to adopt agile practices easily for offshore development.

III. USE OF PATTERNS IN AGILE ADOPTION

This section first explains how patterns are being used to help companies adopting agile practices, and then presents the distributed agile patterns.

A. CURRENT PATTERNS FOR AGILE ADOPTION

The term "pattern" is commonly referred to as a reusable solution for a recurring problem within a given context [49]. A pattern usually consists of four parts, which are pattern name, problem, solution and consequence. The mostly used types of patterns which cover the whole software development life cycle: Requirement patterns [50], Analysis patterns [51], Design patterns [49], Architecture patterns [52], Idioms, and Anti-Patterns [53].

Based on the above definition of patterns, agile patterns are defined as "focus on how an agile practice is being repeatedly modified and used in order to solve a recurring agile problem in a particular context" [54]. To clarify the difference between what is considered as an agile practice and an agile pattern; an agile practice consists of any agile method and technique, which helps, in the application of agile methodology however agile patterns consist of recurring best agile practices, which are being used for the application of agile methodology.

As companies started to adopt offshoring, new patterns were being designed. Noll *et al.* [55] designed patterns for decision support systems that practitioners can use in offshore development. Lescher [56] in Siemens designed collaboration patterns for offshoring. His pattern catalogue consists of five patterns, which focused on the improvement of the communication and coordination among distributed teams. For example he identified one pattern Tailored Training, that focused on co-locating teams at the start of the project so that they can be trained and familiarised with the whole team and technologies that are needed in the project. Similarly, collaborative patterns were presented by van Heesch [57] which focused on how collaboration can be improved among distributed teams.

Similarly, researchers have worked on designing patterns for offshore agile development. Cordeiro *et al.* [58] designed a solution for offshore development by combining organisational patterns with Scrum. They identified 21 patterns and designed a language to structure patterns. Välimäki [59] focused on providing patterns for management techniques in offshore scrum projects.

B. DISTRIBUTED AGILE PATTERNS

So far the work done in this area is either generic or too focused on management and coordination of offshore projects. In contrast, this research identified a catalogue of distributed agile patterns, which will aid practitioners in adopting agile practices in offshoring context. While studying many cases from literature it was observed that many companies used common agile practices to overcome offshore challenges [60]–[63]. In other studies [56], [57], [64], [65], they designed communication and coordination patterns to overcome communication and coordination challenges that onshore and offshore team members face while working together. In [66], they designed patterns for project management.

From the observation it was established that recurring offshore problems where being solved by recurring agile practices. Based on the definition of agile patterns, **distributed agile pattern** is being defined as ‘‘the adaptation of an agile practice that is being repeatedly applied in order to solve recurring challenges in a distributed project scenario’’ [54]. In this research, systematic literature review and content analysis research methods were used to design the distributed agile patterns catalogue. In the Table 3 a summary of the existing techniques from the literature are presented which are being used for offshore development, we have compared them with our distributed agile patterns catalogue.

TABLE 3. Comparison of distributed agile patterns with existing literature.

No.	Article	Contribution	Distributed Agile Pattern
1	Hofmann [67]	Proposed a framework to offer convincing trainings, learning factories and hardware solutions for good training of distributed part-time teams in offshore agile development.	Distributed agile pattern catalogue identifies 4 categories of distributed agile challenges and provides a catalogue to practitioners that is based on the how agile is being adopted and used in offshore scenario.
2	Khan [68]	Proposed a framework for evaluating communication and coordination process in offshore development using fuzzy logic. Identified 6 critical challenges and 75 practices.	
3	Aggarwal [69]	In order to shorten release time of a product, they used product line engineering for their offshore agile projects. The paper mostly focused on release cycles.	
4	Mahajan [70]	Proposed an algorithm for distributed agile development processes to lower the risk profile based on cost and program.	
5	Licorish [71]	Conducted a study on IBM Rational Jazz practitioners to understand how self-organising teams work in distributed agile project.	

IV. RESEARCH METHOD

The research has been conducted by carrying out Systematic Literature Review following the guidelines of Kitchenham [72]. SLR was used to select primary studies from the existing literature. Krippendorff [73] was also used for content analysis on the semi-structure interviews that were conducted to verify the results. After the identification of the distributed agile patterns, for verification and validation reflection workshop proposed by Kerth [74] was conducted. The objective of this study is to provide the answers to the following questions:

Research Question: What are the recurring adaptions of agile practices used in Offshore software development companies?

To be more specific, the study focused on the following two sub-questions:

Research Question 1: What agile practices are commonly used to handle offshore software development challenges?

Research Question 2: Are those practices being used to solve a recurring offshore software development problem that can be considered patterns?

A. DATA SOURCE AND SEARCH STRATEGIES

All the papers searched were written in English and were available online. The search strategy consisted of searching electronic databases as well as manually searching of conference proceedings. The electronic databases used are: IEEEXplore, ACM Digital Library, Google Scholar, Elsevier Science, AIS eLibrary, SpringerLink and Taylor Francis Online. The Figure 2 shows the paper review process and how many papers were identified at each stage.

In the first stage, the search items mentioned in Table 4 were used to select the databases. Category 1 has variations of the term ‘‘Global Software Development’’ and Category 2 has variations of the terms used for ‘‘agile practices’’. The combination of all the search items using a Boolean ‘‘AND’’ operator was used, which meant that an articles will only be considered as part of this research if it has both keywords or variations of the keywords, Agile

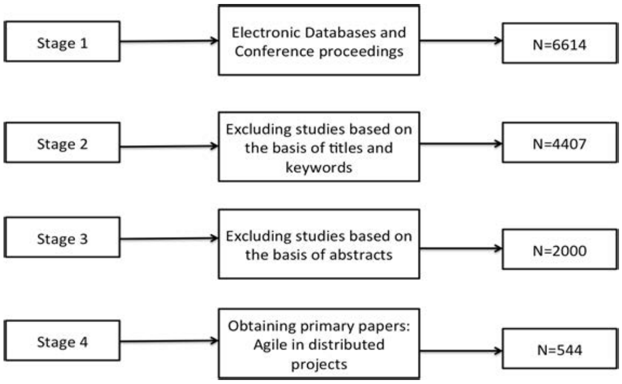

Stage 1
Electronic Databases and Conference proceedings
N=6614
Stage 2
Excluding studies based on the basis of titles and keywords
N=4407
Stage 3
Excluding studies based on the basis of abstracts
N=2000
Stage 4
Obtaining primary papers: Agile in distributed projects
N=544

FIGURE 2. Primary papers selection process.

TABLE 4. Search terms used for this review.

Type	Category	Keywords
1	Global Software Development	Global Software Engineering Global Software Development Offshore Software Development Offshoring Distributed Software Development Distributed Development
2	Use of Agile Practices Agile	Agile practices Agile methods Scrum XP

and Global Software Development. As a result every combination was searched from Category 1 AND Category 2. Excluded articles consisted of editorials, article reviews, prefaces, news, discussion comments and tutorials summary, workshops, poster sessions and panels. The search resulted in finding 6614 articles. The following screening criteria was used to ensure that the papers selected addressed the research problem. "Lesson learnt" reports were also considered in the studies that were based on the opinions of experts to focus on how in offshore projects agile practices are used.

In the second stage studies were excluded based on the relevancy of titles and keywords based on the search items mentioned in Table 4. All the titles were read and 4407 articles were selected. In some cases, the titles failed to clearly identify whether the study was within the scope of this review. In such cases, the articles were included in the next review stage. In the third stage, all the articles that did not focus on offshore development were excluded. Some abstracts were misleading, that they gave little information about the content of the paper or did not state if the study was relevant. Therefore, at this stage, we only included studies that showed relevance to offshore experience, 2000 articles were shortlisted. Based on these points shortlisted papers helped in understanding the current agile practices and if those practices are being repeatedly used to solve offshore problems. Further each criterion was graded on dichotomous scale that is either "yes" or "no". In the final stage, the following criteria was selected in order to obtain primary articles:

1) Is the paper based on research?
2) Are the research aims clear?
3) Is there a satisfactory context to the research?

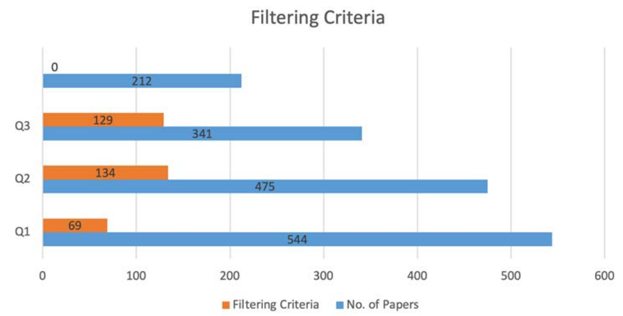


FIGURE 3. Literature filtration based on screening criteria.

4) Was the research design appropriate for addressing the aims of the research?
5) Was data collected as addressed in the research issue?
6) Was the data analysis sufficient?
7) Was the relationship between the researcher and the participants been considered?
8) Are the findings clear?
9) Does the study have value for research/practice?

Based on the above criteria 544 articles were selected. Out of 544 articles, 212 papers were selected based on the screening criteria, Figure 3 shows the filtration of papers based on the questions:

1) Does a paper address any challenge that occurs in the application of agile practices in offshore development?
2) In distributed projects have they mentioned any real life experience showing the use of agile practices?
3) Has any agile practices been repeatedly used to solve any problems in distributed projects

The selected papers satisfied all three criteria, were graded "yes" in all three. That is, papers that did not show agile practices used to solve recurring problems were excluded, for example if only one paper mentioned that they used split pair programming to improve their coding. For a paper to be considered in this research an agile practice needs to solve an agile challenges and should be repeated in at least two papers to be classified as a distributed agile pattern. Figure 4, shows occurrence of an agile practice in literature such as asynchronous information practices occurred in nearly 200 papers whereas project charter occurred in 11 papers. Based on how many times a practice occurred in the literature, an overall 15 patterns were identified.

In order to verify the identified 15 patterns, 20 semi-structured interviews were conducted, which consisted of nine open-ended questions, which covered different aspects of offshoring. The questionnaire is mentioned in the appendix, which is available at https://bit.ly/3DknJC2. This helped in obtaining experts points of view regarding how agile practices are being used in the industry. The selection of companies was done based on their experience in offshore software development. The companies were categorised into 3 classes, which are startup, break-even and profitable. Companies were categorised as startups that are still in early stages of earning revenue, likewise break-even, are those companies that have reached break-even status in their finances and the companies that are generating profit are categorised as profitable. The description of the companies interviewed is mentioned in Table 5. Based on the search criteria, 15 distributed agile patterns, were identified which were verified using a reflection workshop. Feedback was collected from the companies that were invited during the workshop in order to gather their views on the patterns catalogue. Based on their view the following aspects were assessed:

- How complete the distributed pattern catalogue is?
- How useful this catalogue may be to the practitioners that want to overcome distributed development challenges?

TABLE 5. Detail of companies interviewed.

Category	Experience in offshoring (years)	Offshore Projects	No. of Companies
Startup	1-2	3	4
Break-even	>3	8-9	6
Profitable	>5	>10	10

V. RESULTS

This section presents the initial distributed agile patterns identified from the literature and interviews, and the revised patterns based on the results of the reflection workshop.

A. DISTRIBUTED AGILE PATTERNS IDENTIFIED FROM LITERATURE AND INTERVIEWS

By conducting literature review and interviews the pattern catalogue was developed. In order to document the distributed agile pattern catalogue Gamma's design pattern template was used, as it is perceived as the first pattern catalogue and to preserve familiarity for the practitioners [49]. However, the template was customised to capture the findings. The template contained sections such as pattern name, intent, also known as, category, applicability, participants, collaboration, consequences, known uses, and related patterns. Sections such as structure, implementation and sample code were removed, as they were not relevant to the findings. Based on the findings, 15 distributed agile patterns were identified and are placed into four categories based on the type of problem they solved. The four categories are management, communication, collaboration and verification. Figure 5 shows the mapping of distributed agile patterns onto the Scrum development lifecycle. Asynchronous Information Transfer, Synchronous Communication and Visit Onshore-Offshore are not mapped specifically onto the figure as the offshore company can decide when and how to use them throughout the whole lifecycle. Due to the limited space available for this article, one example of patterns is presented here, whereas the full catalogue is available at the following URL: https://bit.ly/3lqcNwP

1) DISTRIBUTED SCRUM OF SCRUMS PATTERN

In agile methodology, Scrum is an iterative and incremental project management approach that provides a simple framework that " inspect and adapt" [75]. We observed that in offshore projects the onshore and offshore team practices separate scrums in order to develop the project. Based on the observed practice we have designed the following pattern details:

2) PATTERN NAME

Distributed Scrum of Scrums Pattern

3) INTENT

To apply scrum, sub-teams are formed based on location. Each team has its own scrum. Scrum of scrums meetings are arranged to discuss the progress of the project, which is attended by key people.

4) ALSO KNOWN AS

Scrum meeting or Meta Scrum

5) CATEGORY

Management category, as this pattern helps the onshore and offshore team manage their separate scrums and keep each other updated of the project progress.

6) MOTIVATION

The motivation of this pattern is to address the communication and coordination, and knowledge transfer challenges. For example consider a team that is divided into sub-teams based on location and they are working on different tasks of a project. It is difficult to have both onshore and offshore team work on the same scrum as they both work on different time zones so in order to work on the same project, both teams work on separate scrums.

To coordinate work both teams arrange a scrum of scrums meeting, which is attended by key people from both teams to update each other of the progress of the project.

7) APPLICABILITY

Use Distributed Scrum of Scrums when:

- Team is distributed over different time zones.
- There is less overlap over working hours between the onshore and offshore teams.

8) PARTICIPANTS

- Distributed onshore and offshore agile team.
- Scrum Masters of agile sub-teams and Product owner.

9) COLLABORATION

- Key members from onshore and offshore teams decide time for Scrum of Scrums meeting.

10) CONSEQUENCES

The Distributed Scrum of Scrums pattern has the following benefits and liabilities:

1) It prevents the onshore and offshore team from wasting time on collaborating tasks with each other through

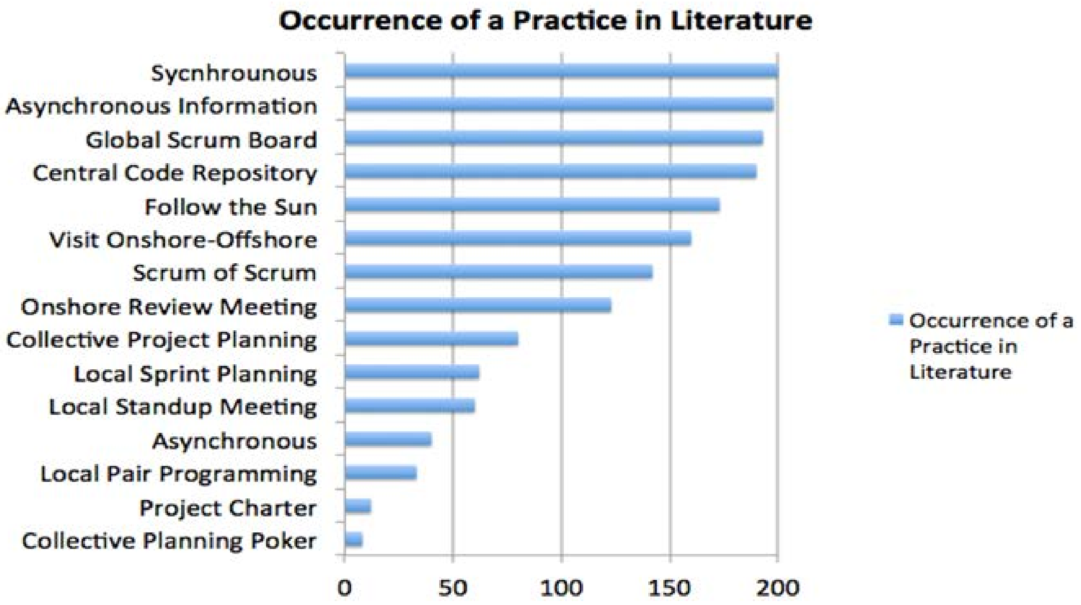


FIGURE 4. Recurring agile practices in the literature.

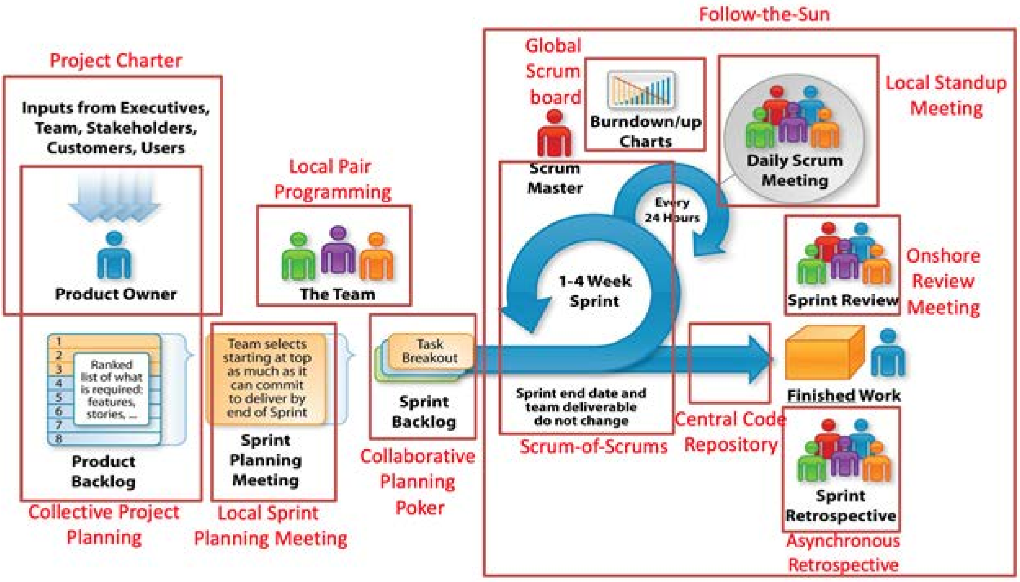


FIGURE 5. Distributed agile patterns application of software development lifecycle.

online tools as both the teams are working on their own scrums so they don't have to wait for each other to complete tasks. This helps overcome the communication and coordination challenges.

2) It provides control to both onshore and offshore team to work on their scrum, which avoids the offshore team from having to adjust working hours based on onshore teams availability. This helps overcome the communication and coordination challenges.
3) It allows key people such as Scrum Masters and Product owners to discuss the progress of the project without having the whole team present which keeps the meeting time boxed and helps in knowledge transfer among the teams.
4) Its limitation is that due to minimum collaboration between the onshore and offshore team, both sub-teams don't feel they are one team.
5) Since only key people attend the Scrum of Scrums meeting, it limits face-to-face interaction of both onshore and offshore team, which affects trust building between both teams.

11) KNOWN USES

CheckFree, used Scrum of Scrums for their work in an Indian offshore firm [76]. Similarly, for their USA, Europe and India offshore teams, Siemens also used Scrum of Scrums [77].

12) RELATED PATTERNS

Distributed Scrum of Scrums pattern is often used with Local Sprint Planning Pattern and Asynchronous Retrospective meeting Pattern as teams on different locations are working on separate story cards resulting in conducting separate retrospective meetings.

Table 6 represents the summary of the ''known uses'' of the patterns by practitioners. This shows how practitioners have used these patterns in real project scenarios such as CheckFree has used Scrum of Scrums and Local Sprint Planning for their offshore projects. Similarly, Yahoo! used Follow-the-sun approach for their offshore teams in which each team worked according to their time zones and synced work using synchronous and asynchronous tools.

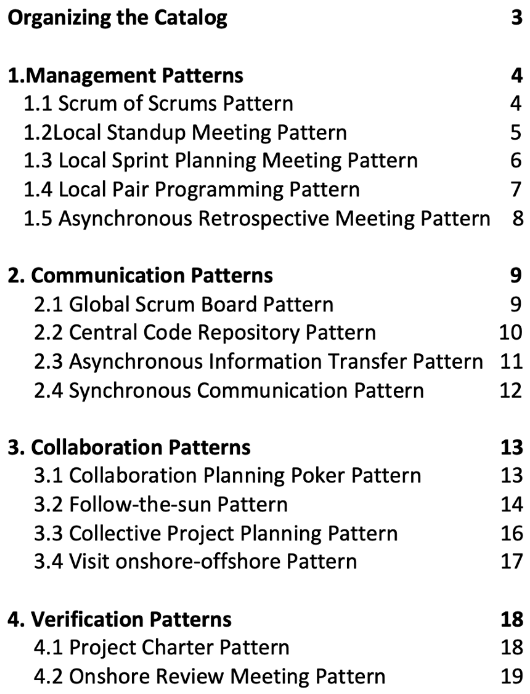
B. REVISED DISTRIBUTED AGILE PATTERNS

In literature there are many approaches through which you can assess results such as using fuzzy approach [99], structural modeling approach [100]. In this research to verify and validate the pattern catalogue, Kerth's [74] reflection workshop method ''The keep/try reflection workshop'' was used. This method was selected to get opinions from experts regarding the correctness of the patterns and if the catalogue is useful to practitioners who want to apply agile to their offshore projects. Lastly we conducted a questionnaire survey as suggested by Sikandar [100] to summarise the patterns from the practitioner's perspective.

For the workshop four companies were invited. Based on our earlier definition of the startup, break-even and profitable we invited 1 startup, 2 break-even and 1 profitable companies. All the companies had at least 2 years experience with agile offshore development. Details of the companies are provided in Table 7.

In order to get both a managerial and development view two participants from each company were invited. All of the participants had experience in distributed agile development. Details of participants are shown in Table 8. The workshop was held in the boardroom of C1 and lasted for seven hours that included presenting the pattern catalogue and collecting feedback from the participants on the flip chart. Table 9 shows the agenda of the workshop. Table 10 shows the format of the flip chart on which feedback was collected. It was based on the reflection workshop of Kerth. The 'Keep these' section refers to all the patterns the participants do not want to change, 'Problems' refers to the patterns they want some changes in and 'Try these' are recommendations that they want us to consider.

The catalogue as in Table 6 was presented in a printed hard form and given to each participant so that they may read it and make notes on it. As the participants were not familiar with the catalogue, one pattern was presented at a time and participants discussed it with each other before moving onto the next pattern. Similarly flip charts were given to the participants so that they can document their opinions. The figure 6 shows the table of contents of the document given to the participants. Each pattern was numbered so that the participants can use them to refer patterns.



Organizing the Catalog	**3**
1.Management Patterns	**4**
1.1 Scrum of Scrums Pattern	4
1.2Local Standup Meeting Pattern	5
1.3 Local Sprint Planning Meeting Pattern	6
1.4 Local Pair Programming Pattern	7
1.5 Asynchronous Retrospective Meeting Pattern	8
2. Communication Patterns	**9**
2.1 Global Scrum Board Pattern	9
2.2 Central Code Repository Pattern	10
2.3 Asynchronous Information Transfer Pattern	11
2.4 Synchronous Communication Pattern	12
3. Collaboration Patterns	**13**
3.1 Collaboration Planning Poker Pattern	13
3.2 Follow-the-sun Pattern	14
3.3 Collective Project Planning Pattern	16
3.4 Visit onshore-offshore Pattern	17
4. Verification Patterns	**18**
4.1 Project Charter Pattern	18
4.2 Onshore Review Meeting Pattern	19

FIGURE 6. Document presented the reflection workshop.

Table 11 shows an example of how we documented the reflection workshop on a flip chart of Company 3 Participant 5 (C3P5).

In the flip chart, C3P5 wants the section 1.2-1.5 to stay the same as presented so they wrote them in the **Keep these** section. In the **Problems** section they want us to change the Central Code Repository pattern as according to them it is currently too generic. In the **Try these** section they have made some suggestions such as to rename the pattern Scrum of Scrums as it causes confusion with the general term used for scrum of scrums practice so they suggested to use the name of Distributed Scrum of Scrums. A summary of all the flip charts is presented in Table 12. From the summary, it can be seen that most of the companies agreed to keep sections 1.2-1.5 the same way as they were presented and similarly, for sections 2.1,2.3-2.4, 3.3-3.4 and 4.1-4.2. However problems were identified in patterns 2.2, 3.1 and 3.2 which were modified based on the suggestions given by the companies. Lastly, there were a few things they wanted to try with patterns 1.1 and 4.2 which were considered in the revised version of the catalogue.

After discussing the pattern catalogue the discussion moved towards how useful the catalogue is in answering the offshore challenges. Table 13 shows, which pattern helps solve the offshore challenges such as Distributed Scrum of Scrums solves socio-cultural and communication and coordination challenges, below we have explained all the patterns and the challenges they solve.

- **Follow the sun pattern**: was agreed by 50% of the participants that it will help improve the communication

TABLE 6. Detail of known uses of distributed agile patterns.

No.	Pattern name	Adapted Practices
1	Distributed Scrum of Scrums Pattern	The traditional practice is that the whole agile team follows a single Scrum however companies that opt offshore development have been seen to adapt this practice by following Distributed Scrum of Scrums: Below are some examples: • CheckFree, used Scrum of Scrums for their work in an Indian offshore firm [76]. • Siemens for their USA, Europe and India offshore teams also used Scrum of Scrums [77].
2	Local Stand-up Meeting	In traditional agile the whole team was present in the daily stand-up meeting but in offshore development they adapted the practice to local stand-up meetings so that each team can conduct their separate meeting. Below are some examples: • Organisations such as PulpCo [78] and Wipro Technologies [79].
3	Local Sprint Planning Meeting Pattern	In traditional agile there would be one sprint planning meeting but in offshore development the adapted practice is that each remote team has its own Sprint Planning meeting. Below is an example: • To plan sprint activities CheckFree used local sprint planning meetings for their offshore projects [76].
4	Local Pair Programming Pattern	Since the teams are working on separate Sprints, the adapted agile practice in offshore is that teams form local programming pairs. Below is an example: • An organisation with offshore offices in India, U.S West Coast, U.S Mid-West and U.S East Coast created pairs based on physical locations [80].
5	Asynchronous Retrospective Meetings Pattern	Traditionally the whole team would be at one location so Retrospective meetings were held at one location but in adapted offshore agile practice each team conducts their own retrospective meetings and share meeting minutes with each other. Below is an example: • Elastic Path, partner used asynchronous retrospective sessions when they offshored their work to Luxsoft [81].
6	Global Scrum Board Pattern	As the team is distributed, the offshore agile adapted practice is that the teams maintain a Global Scrum Board so that they can view each others progress. Below are some examples: • FAST with headquarters in Norway used a globally shared Scrum board [82]. • Similarly an organisation that had offshore offices across North America, South America and Asia used global scrum board to improve the productivity of their offshore teams [83]. • Companies like Valtech [84], Telco [85], BNP Paribas [16], Aginitys LLC [86] and SirsiDynix [87] also use global scrum boards for their teams.
7	Central Code Repository Pattern	Similarly the agile offshore teams maintain a central code repository across remote teams. Below are some examples: • WDSGlobal has development centers located in UK, USA and Singapore. In order to minimize duplications and maintenance cost they shared code in a central repository [88]. • Central code repository is used for many distributed projects such as Extol International [89], Valtech [84], Manco [85], Aginity LLC [86], SirsiDynix [87], CEInformant [90] and ABC Bank [91].
8	Asynchronous Information Transfer Pattern	Traditionally agile encourages face-to-face meetings, however the adapted offshore practice is to use asynchronous tools to communicate with each other due to time differences. Below are some examples: • Research conducted by VTT Technical Research Centre of Finland and National University of Ireland showed that for storing documents and meeting minutes they used online tools and that for decisions and queries they used emails [92]. • Twiki was used by Valtech for asynchronous communication [84].
9	Synchronous Communication Pattern	Similarly in offshore agile development teams highly depend on synchronous tools for day-to-day information exchange. Below is an example: • For their offshore suppliers in India and Romania, CampusSoft used synchronous communication [93].
10	Collaborative Planning Poker Pattern	Even though the adapted practices are to have separate Scrum and Sprints it has been observed that offshore agile teams prefer to have a collaborative planning poker activity as it helps the team decide the work estimations. Below is an example: • To estimate story cards for their offshore centers across America and Asia, UShardware used planning poker activity [94].
11	Follow-the-Sun Pattern	Traditionally as the whole team was located at one location the working hours were fixed, but the adapted offshore agile practice is that each remote team follows there own office timing instead of synching with parent organization. Below are some examples: • Yahoo! used follow-the-sun approach when they offshored their Yahoo! Podcast product [95]. • Similarly the follow-the-sun approach is used by organisations such as WDSGlobal [88] and Wipro Technologies [79].
12	Collective Project Planning Pattern	Like Planning Poker in offshore adapted practices it has been observed that teams prefer collective project planning as it helps in establishing project timeline. Below are some examples: • To do collective project planning, FAST got their team together for this activity [82]. • Similarly Siemens also used collaborative project planning [26], [77].
13	Visit Onshore-Offshore Team Pattern	In adapted offshore agile, it has been observed that teams that visit each other establish trust which is important for team building. Below is an example: • Ericsson for their XaaS platform development, used visit onshore-offshore practices [96].
14	Project Charter Pattern	Project Charter is not a part of traditional agile but it has been observed that in offshore projects, teams prefer to write a project charter in order to have clear objectives. Below are some examples: • For their offshore projects, IONA Technologies used Project Charter [97]. • Similarly during Agile-at-Scale Delivery project charter was used [98].
15	Onshore Review Meeting Pattern	Despite the team is distributed, the adapted offshore agile practice is that the review meeting will be conducted at the onshore location. Below are some examples: • Onshore review meetings were used by Wipro Technologies so that they could get quick feedback [79]. • SirsiDynix used onshore review meetings to demo their work [87].

TABLE 7. Invited companies details.

Company	Category	Type of Business	HQ	Location of Business	Experience in Agile
C1	Startup	IT services	Pakistan	Dubai, UK	2 years
C2	Breakeven	Product-based	Pakistan	USA, UK	5 years
C3	Breakeven	Multinational IT service provider	Pakistan	Dubai	8 years
C4	Profitable	Software solutions	USA	Pakistan	10 years

TABLE 8. Details of the reflection workshop participants.

Company	Participant Name	Job Title	Role in Agile	Exp. in Agile (years)	Exp. in DAP
C1	P1	CEO	Scrum Master	2	2
	P2	Developer	Developer	2	1
C2	P3	CEO	Product Owner	5	5
	P4	Senior Developer	Developer	5	5
C3	P5	CEO	Scrum Master	8	8
	P6	Developer	Developer	5	4
C4	P7	Senior developer	Developer	8	5
	P8	Senior Software Engineer	Developer	8	4

TABLE 9. Agenda of reflection workshop.

No.	Agenda items	Time
1	Tea and Networking	0930-1000 Hrs
2	Introduction to reflection workshop and how it will be conducted	1000-1100 Hrs
3	Presented each Distributed Agile Pattern and discussed them one by one.	1100–1400 Hrs
4	Lunch	1400-1500 Hrs
5	Feedback and recommendation	1500-1600 Hrs

TABLE 10. Flip chart format for the reflection workshop.

Keep these	Try these
Problems	

TABLE 11. Flip chart of company 3 participant 5 (C3P5).

Keep these	Try these
• Keep 1.2,1.3,1.4, 1.5 • Keep 2.2,2.3,2.4 • Keep 3.1,3.2,3.4 • Keep 4	• 1.1: Rename the pattern in order to avoid confusion. • 3.3: Let the distributed team decide how they want to collaborate
Problems	
• 2.2: Pattern needs to be changed as its too generic.	

and coordination among the distributed team members. However the remaining 50% questioned the practice as by following the sun pattern the overlapping working hours reduces which may result in less real time communication among the distributed teams.

- **Asynchronous retrospective meetings**: was agreed by 65% participants that it will help improve communication and coordination gap as well as aid in knowledge transfer as teams can conduct their retrospective and share meeting minutes with offshore teams.

TABLE 12. Companies flip chart summary.

Keep these	Try these
• Keep 1.2, 1.3,1.4,1.5 • Keep 2.1, 2.3,2.4 • Keep 3.3, 3.4 • Keep 4.1,4.2	• 1.1: Rename pattern for clarity. • 1.1: Maybe make patterns 1.1-1.5 sub-parts of each other as they are related. • 4.2: Let offshore team sometimes be part.
Problems	
• 2.2: Too generic, needs to be modified. • 3.1: Onshore as well as offshore team members should be part of planning poker. • 3.2: Limited applicability due to time zones.	

- **Project charter pattern**: 75% of the participants agreed that this pattern could help resolve trust issues as well as aid in bridging the communication and coordination gaps and solve issues of knowledge transfer. However 25% still believed that a single document could not help establish trust let alone solve issues of communication and coordination.
- **Distributed scrum of scrums**: 80% participants agreed that it will improve communication and coordination and will also solve knowledge transfer issues in distributed teams. As in this pattern, teams are allowed to conduct their own separate scrums and just have to share updates synchronously.
- **Local stand-up meeting**: 80% participants agreed that local stand-up meeting aids in solving communication and coordination issues, it also solves knowledge transfer issues. Teams can independently conduct their stand-up meeting and share key points with relevant stakeholders.

- **Local sprint planning meeting**: It was agreed by 80% of the participants that local sprint planning meetings issues in communication and coordination as each sub team can plan their independent sprints and share knowledge using tools with the whole team to track progress of the project.
- **Local pair programming**: 80% of the participants agreed that pairs should be formed based on location as it eases in communication and coordination among the team members.
- **Onshore review meeting**: Since the team members are distributed, 75% of the participants agreed that onshore review meetings help solve communication and coordination issue as the onshore team can directly present the work to client and the feedback can be shared with the offshore team which facilitates knowledge sharing
- **Asynchronous information transfer**: As the teams are distributed they highly depend on asynchronous tools for information sharing and communication and coordination. 80% of the participants agreed to its applicability.
- **Collective project planning**: 100% of the participants agree on collective project planning to solve all four challenges as when the team is co-located they can work together as one team and avoid all the challenges due to offshoring.
- **Collaborative planning poker**: When the team members will be co-located and working on estimation together, 100% participants agree that this pattern will solve all four challenges.
- **Central code repository**: 100% of the participants agreed that having a central code repository will solve communication and coordination issues and will streamline knowledge transfer of code among teams.
- **Synchronous Communication**: When the team members are distributed they need to rely heavily on synchronous tools to share knowledge, hence 100% of the participants agree on synchronous communication.
- **Global scrum board**: All participants agree that global scrum board is a key to solving all the challenges as having a centralised board showing all the information solves communication and coordination issues as well as knowledge transfer. Since everyone can view and edit the board, it creates trust and encourages socio-cultural interactions.
- **Visit onshore-offshore team pattern**: 100% participants agreed that this is a very useful pattern as it helps in establishing trust among the distributed teams. This pattern in turn also aids communication, coordination and knowledge sharing among the members.

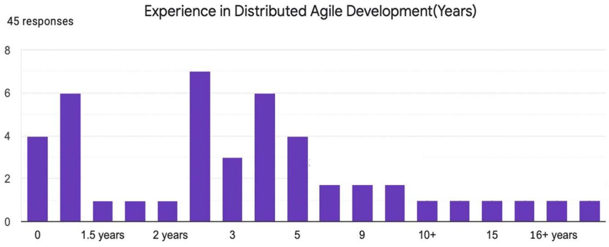


FIGURE 7. Distributed agile experience of questionnaire participants.

After discussing the effectiveness of each pattern in the application of agile in offshore projects feedback was collected based on how useful the catalogue is for practitioners. Based on the Distributed Agile Patterns catalogue we designed a questionnaire to determine the usefulness of the patterns, which is available at https://bit.ly/3CYSFs1.The questionnaire was distributed online using Google Forms. The link was shared through email and LinkedIn. A total of 45 practitioners participated. An overview of their distributed agile experience is mentioned in Figure 7. Usefulness of the Distributed Agile Pattern catalogue was identified through empirical study shown in Table 14. In this table, the EA (Extremely Agree), MA (Moderately Agree), SA (Slightly Agree), SDA (Slightly Disagree), MDA (Moderately Disagree) and EDA (Extremely Disagree) show 7 levels of the participant's agreement to the questions. It has been divided into two main columns, pattern name and expert observation. The ''pattern name'' lists the names of all the patterns in the catalogue and expert column records experts experience about the usefulness of each pattern which is further divided into three columns i.e. ''Positive'', '' Negative'' and ''Neutral''. For the purpose of analysis we grouped the responses into three groups X, Y, and Z. The X group counts the frequency of all the positive responses such as (slightly agree, moderately agree, and extremely agree), group Y consists of neutral responses and group Z consists of negative responses such as (extremely disagree, moderately disagree, and slightly disagree). By negative impact we mean that the practitioner did not agree that the pattern was useful for solving distributed agile challenges. We suggest that organisations that want to use agile for their offshore software projects should use our pattern catalogue. Analysing the percentage values of the ''Negative'' column in the Table, we can see that most of the values are below 18%. This shows that majority of the experts agree that the patterns are useful for agile offshore development.

To find the usefulness of a pattern the criteria used was that if a pattern is answered as agree in the questionnaire with a percentage of more than or equal to 50% then we consider that the pattern is useful to solve a distributed agile challenge. This research criteria have been used for similar research [100], [101].

C. ANSWERING RESEARCH QUESTIONS

As in this research the main focus was to answer the following research questions.

1) COMMONLY USED AGILE PRACTICES TO HANDLE OFFSHORE SOFTWARE DEVELOPMENT (RQ1)

Based on the selected 212 papers from literature 15 commonly used agile practices from literature were identified.

TABLE 13. Distributed agile patterns challenges solved.

Pattern name	Trust	Socio-Cultural	Communication and Coordination	Knowledge Transfer	Agreed
Distributed Scrum of Scrums			✓	✓	80%
Local Standup Meeting			✓	✓	80%
Follow-the-Sun			✓	✓	50%
Onshore Review Meeting			✓	✓	75%
Collective Project Planning	✓	✓	✓	✓	100%
Project Charter	✓		✓	✓	75%
Collaborative Planning Poker	✓	✓	✓	✓	100%
Global Scrum Board	✓	✓	✓	✓	100%
Local Sprint Planning Meeting			✓	✓	80%
Local Pair Programming			✓		80%
Central Code Repository			✓	✓	100%
Asynchronous Retrospective Meetings			✓	✓	65%
Asynchronous Information Transfer			✓	✓	80%
Synchronous Communication	✓	✓	✓	✓	100%
Visit Onshore-Offshore Team	✓	✓	✓	✓	100%

TABLE 14. Challenges identified in offshore software development.

No.	Pattern Name	Experts Observation($n = 45$)											
		Positive (P)					Neutral(N)		Negative(N)				
		EA	MA	SA	X	%	Y	%	SDA	MDA	EDA	Z	%
1.	Distributed Scrum of Scrums	20	12	2	34	76%	3	7%	5	2	1	8	18%
2.	Local Standup Meeting	17	20	6	43	96%	1	2%	1	0	0	1	2%
3.	Follow-the-Sun	9	22	7	38	84%	4	9%	3	0	0	3	7%
4.	Onshore Review Meeting	30	7	3	40	89%	3	7%	1	1	0	2	4%
5.	Collective Project Planning	13	16	4	33	74%	6	13%	3	3	0	6	13%
6.	Project Charter	12	23	5	40	89%	2	4%	2	1	0	3	7%
7.	Collaborative Planning Poker	15	16	10	41	91%	3	7%	1	0	0	1	2%
8.	Global Scrum Board	20	20	2	42	93%	3	7%	0	0	0	0	0%
9.	Local Sprint Planning Meeting	18	22	3	43	96%	1	2%	1	0	0	1	2%
10.	Local Pair Programming	8	18	10	36	80%	7	16%	2	0	0	2	4%
11.	Central Code Repository	31	13	0	44	98%	1	2%	0	0	0	0	0%
12.	Asynchronous Retrospective Meetings	8	16	10	34	76%	3	7%	5	1	2	8	17%
13.	Asynchronous Information Transfer	16	10	8	34	76%	6	13%	2	1	2	5	11%
14.	Synchronous Communication	18	22	3	43	96%	2	45	0	0	0	0	0%
15.	Visit Onshore-Offshore Team	18	15	5	38	84%	4	9%	1	2	0	3	7%

These identified practices were observed to have been modified for the adoption of distributed agile development. This helped in answering the first research question in which commonly used agile practices in offshore are adapted to deal with offshore software development. With the help of literature review 15 practices that are being used to solve issues in offshore development were identified.

2) RECURRING AGILE PRACTICES USED FOR OFFSHORE SOFTWARE DEVELOPMENT PROBLEM (RQ2)

As we answered the RQ1 by identifying commonly used agile practices in solving offshore development issues, the next step was to determine if those practices recurred in solving agile problems. The Figure 4 shows how many times in literature the 15 agile practices recurred in solving an offshore problem. Since these modified practices recurred to solve agile problems we classified them as distributed agile patterns. Based on these recurring practices the distributed agile patterns catalogue was designed. Table 15 shows the categories of the distributed agile patterns catalogue.

VI. THREAT TO VALIDITY

There are four main types of threats to validity, which are internal validity, external validity, construct validity and conclusion validity. We conducted threat to validity for our results.

While identifying the key inherent challenges of offshoring the research focused only on the broad categories of the challenges such as Trust, Communication & Coordination, Socio-Cultural and Information Transfer. We did not focus on challenges that did not lie in these categories as their occurrence in literature was not significant. This can pose a threat to validity as their maybe new challenges that does not fall into these categories.

For the systematic literature review the selection of the studies were chosen based on a set of specific keywords, which can pose as a threat to validity as some studies could have been, eliminated as the keywords did not cover them. The screening process was done manually by reading the 544 papers and selecting 212 based on the papers fulfilling the three question screening criteria, since this was done manually it can pose a threat to validity as the understanding of researchers about papers can vary and they could have made an error and also there is a chance of bias.

For interviews and workshop conducted in this study limited number of participants were selected for the sample set. This can pose as a threat to validity as the selection of research participants was limited by the willingness of experts to participate and the availability of experts in the domain of agile. As with any empirical software engineering project, a very high number of variables can affect the project.

TABLE 15. Distributed agile patterns categories.

Management Patterns	Communication Patterns	Collaboration Patterns	Verification Patterns
• Scrum of Scrums • Local Standup meeting • Local Sprint Planning • Local Pair Programming • Asynchronous Retrospective	• Global Scrum Board • Central Code Repository • Asynchronous Information Transfer • Synchronous Communication	• Collaborative Planning Poker • Follow-the-sun • Collective Project Planning • Visit onshore-offshore	• Project Charter • Onshore Review Meeting

Similarly, for the distributed agile patterns catalogue it is difficult to identify any one factor that can lead to the success and failure of the catalogue. However, the usefulness of the catalogue was clearly evident from the feedback of the participants.

Patterns generally refer to generalised solutions within a given context. We do not claim the catalogue is comprehensive. We want to encourage researchers to identify more recurring agile practices that are being used in order to overcome offshore challenges.

VII. CONCLUSION AND FUTURE WORK

The adoption of agile practices in offshoring is not a straightforward process. The 15 distributed agile patterns catalogue based on the literature and semi-structured interviews was designed to help the practitioners who want to adopt agile for their offshore projects. The catalogue was validated using Kerth's reflection workshop method and the usefulness of the catalogue was determined through questionnaire that was filled by 45 industrial participants. Based on the observation it can be seen that the negative responses were only less than 18%. Practitioners can use this catalogue to understand how agile practices are affected in a distributed scenario and can apply these patterns for when they want to adopt agile practices. The generalisation of the patterns helps practitioners in adapting them for their own projects.

Future work is planned which includes to conduct a case study on the application of the pattern catalogue adoption and to design a tool to help aid users in deciding which pattern suits their requirements and to replicate this study in distributed teams of the most varied types of companies. For upcoming work, we suggest to study how agile methodologies have evolved and the reason why they have evolved to measure whether these adaptations have delivered better results.

REFERENCES

[1] C. Jahns, E. Hartmann, and L. Bals, "Offshoring: Dimensions and diffusion of a new business concept," *J. Purchasing Supply Manage.*, vol. 12, no. 4, pp. 218–231, Jul. 2006.

[2] A. Van Zoest, "Offshoring practices in the UK—Where are the limits?" Inst. Public Policy Res., London, U.K., Background Paper, 2004.

[3] C. A. Garner, "Offshoring in the service sector: Economic impact and policy issues," *Econ. Review-Federal Reserve Bank Kansas City*, vol. 89, pp. 5–38, Jun. 2004.

[4] L. Pilatti and J. L. Nicolas Audy, "Global software development offshore insourcing organizations characteristics: Lessons learned from a case study," in *Proc. IEEE Int. Conf. Global Softw. Eng. (ICGSE)*, Oct. 2006, pp. 249–250.

[5] (2013). *Global Wages Comparison*. Accessed: Oct. 28, 2013. [Online]. Available: https://wageindicator.co.U.K./

[6] M. Kobayashi-Hillary, *Outsourcing to India: The Offshore Advantage*. Berlin, Germany: Springer, 2005.

[7] D. Èmite and C. Wohlin, "A whisper of evidence in global software engineering," *IEEE Softw.*, vol. 28, no. 4, pp. 15–18, Jul. 2011.

[8] J. D. Herbsleb and D. Moitra, "Global software development," *IEEE Softw.*, vol. 18, no. 2, pp. 16–20, Mar. 2001.

[9] D. Damian and D. Moitra, "Guest Editors' introduction: Global software development: How far have we come?" *IEEE Softw.*, vol. 23, no. 5, pp. 17–19, Sep. 2006.

[10] *Manifesto for Agile Software Development*. Accessed: Jan. 20, 2020. [Online]. Available: http://agilemanifesto.org

[11] P. Abrahamsson, J. Warsta, M. T. Siponen, and J. Ronkainen, *New Directions on Agile Methods: A Comparative Analysis*. Piscataway, NJ, USA: IEEE Computer Society, May 2003, pp. 244–254.

[12] R. Hoda, N. Salleh, and J. Grundy, "The rise and evolution of agile software development," *IEEE Softw.*, vol. 35, no. 5, pp. 58–63, Sep. 2018.

[13] T. H. Muhammad Faisal Nisar, "Agile methods handling offshore software development issues," in *Proc. 8th Int. Multitopic Conf. (INMIC)*, Dec. 2004, pp. 417–422.

[14] M. Simons, "Internationally agile," in *InformIT*. Mar. 2002. [Online]. Available: https://www.informit.com/articles/article.aspx?p=25929

[15] L. Hayes, "Everything you know about offshore outsourcing is wrong," in *Datamation Magazine*. USA: Datamation Magazine, 2003.

[16] V. Massol, *JUint in Action*, 1st ed. London, U.K.: Pearson, Nov. 2003.

[17] I. Ghani, A. Lim, M. Hasnain, I. Ghani, and M. I. Babar, "Challenges in distributed agile software development environment: A systematic literature review," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 9, pp. 4555–4571, 2019.

[18] P. S. Taylor, D. Greer, P. Sage, G. Coleman, K. McDaid, and F. Keenan, "Do agile GSD experience reports help the practitioner?" in *Proc. Int. Workshop Global Softw. Develop. Practitioner (GSD)*, New York, NY, USA, 2006, pp. 87–93.

[19] D. Šmite, N. B. Moe, and P. J. Ågerfalk, "Fundamentals of agile distributed software development," in *Agility Across Time and Space*. Berlin, Germany: Springer, 2010, pp. 3–7.

[20] R. Vallon, B. J. da Silva Estácio, R. Prikladnicki, and T. Grechenig, "Systematic literature review on agile practices in global software development," *Inf. Softw. Technol.*, vol. 96, pp. 161–180, Apr. 2018.

[21] R. Camara, A. Alves, I. Monte, and M. Marinho, "Agile global software development: A systematic literature review," in *Proc. 34th Brazilian Symp. Softw. Eng.*, 2020, pp. 31–40.

[22] M. A. Akbar, M. K. Shad, F.-W. Lai, and S. Hussain, "Towards successful agile development process in software outsourcing environment: A systematic literature review," *Int. J. Bus. Innov. Res.*, vol. 23, no. 2, pp. 141–167, 2020.

[23] V. N. Vithana, D. Asirvatham, and M. G. M. Johar, "An empirical study on using agile methods in global software development," in *Proc. 18th Int. Conf. Adv. ICT Emerg. Regions (ICTer)*, Sep. 2018, pp. 150–156.

[24] R. Nardelli, "A comparative analysis of in-house and offshore software development by using agile methodologies at the design/code phase of software development: An empirical study," Ph.D. dissertation, School Comput. Sci. Inf. Syst., Pace Univ., New York, NY, USA, 2019.

[25] D. Šmite, N. B. Moe, and J. Gonzalez-Huerta, "Overcoming cultural barriers to being agile in distributed teams," *Inf. Softw. Technol.*, vol. 138, Oct. 2021, Art. no. 106612.

[26] A. Avritzer and A. Lima, "An empirical approach for the assessment of scheduling risk in a large globally distributed industrial software project," in *Proc. 4th IEEE Int. Conf. Global Softw. Eng.*, Jul. 2009, pp. 341–346.

[27] P. L. Bannerman, E. Hossain, and R. Jeffery, "Scrum practice mitigation of global software development coordination challenges: A distinctive advantage?" in *Proc. 45th Hawaii Int. Conf. Syst. Sci.*, Jan. 2012, pp. 5309–5318.

[28] P. Lous, P. Tell, C. B. Michelsen, Y. Dittrich, and A. Ebdrup, "From scrum to agile: A journey to tackle the challenges of distributed development in an Agile team," in *Proc. Int. Conf. Softw. Syst. Process*, 2018, pp. 11–20.

[29] D. Susan and J. G. Holmes, "The dynamics of interpersonal trust: Resolving uncertainty in the face of risk," *Cooperation and Prosocial Behavior*. New York, NY, USA: Cambridge Univ. Press, 1991, p. 190.

[30] D. M. Rousseau, S. B. Sitkin, R. S. Burt, and C. Camerer, "Not so different after all: A cross-discipline view of trust," *Acad. Manage. Rev.*, vol. 23, no. 3, pp. 393–404, Jul. 1998.

[31] R. Prikladnicki and J. L. N. Audy, "Managing global software engineering: A comparative analysis of offshore outsourcing and the internal offshoring of software development," *Inf. Syst. Manage.*, vol. 29, no. 3, pp. 216–232, Jun. 2012.

[32] R. Sabherwal, "The role of trust in outsourced IS development projects," *Commun. ACM*, vol. 42, no. 2, pp. 80–86, Feb. 1999, doi: 10.1145/293411.293485.

[33] S. Vasudeva Shrivastava and H. Date, "Distributed agile software development: A review," 2010, *arXiv:1006.1955*.

[34] H. Holmstrom, E. Conchuir, P. Agerfalk, and B. Fitzgerald, "Global software development challenges: A case study on temporal, geographical and socio-cultural distance," in *Proc. IEEE Int. Conf. Global Softw. Eng. (ICGSE)*, Oct. 2006, pp. 3–11.

[35] E. Carmel, *Global Software Teams: Collaborating Across Borders and Time Zones*. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

[36] N. Matloff, "Offshoring: What can go wrong?" *IT Prof.*, vol. 7, no. 4, pp. 39–45, Jul. 2005.

[37] F. Lanubile, F. Calefato, and C. Ebert, "Group awareness in global software engineering," *IEEE Softw.*, vol. 30, no. 2, pp. 18–23, Mar. 2013.

[38] Y. I. Alzoubi, A. Q. Gill, and A. Al-Ani, "Empirical studies of geographically distributed agile development communication challenges: A systematic review," *Inf. Manage.*, vol. 53, no. 1, pp. 22–37, Jan. 2016.

[39] R. Qureshi and A. Al-Zaidi, "Global software development geographical distance communication challenges," *Int. Arab J. Inf. Technol.*, vol. 14, no. 2, pp. 215–222, 2017.

[40] K. Conboy and N. Carroll, "Implementing large-scale agile frameworks: Challenges and recommendations," *IEEE Softw.*, vol. 36, no. 2, pp. 44–50, Mar. 2019.

[41] C. Ebert, *Global Software and IT: A Guide to Distributed Development, Projects, and Outsourcing*. Hoboken, NJ, USA: Wiley, 2011.

[42] J. D. Herbsleb, D. J. Paulish, and M. Bass, "Global software development at siemens: Experience from nine projects," in *Proc. 27th Int. Conf. Softw. Eng. (ICSE)*, May 2005, pp. 524–533.

[43] T. H. Davenport and L. Prusak, *Working Knowledge: How Organizations Manage What They Know*. Brighton, MA, USA: Harvard Business Review, 1998.

[44] A. A. Khan, J. Keung, M. Niazi, S. Hussain, and A. Ahmad, "Systematic literature review and empirical investigation of barriers to process improvement in global software development: Client–vendor perspective," *Inf. Softw. Technol.*, vol. 87, pp. 180–205, Jul. 2017.

[45] S. Radoff, "Improved cross-cultural communication increases global sourcing productivity," Accenture, USA, Tech. Rep. #Acc-06-01, May 2006.

[46] M. F. Abrar, M. S. Khan, S. Ali, U. Ali, M. F. Majeed, A. Ali, B. Amin, and N. Rasheed, "Motivators for large-scale agile adoption from management perspective: A systematic literature review," *IEEE Access*, vol. 7, pp. 22660–22674, 2019.

[47] S. Ali and S. U. Khan, "Critical success factors for software outsourcing partnership (SOP): A systematic literature review," in *Proc. IEEE 9th Int. Conf. Global Softw. Eng.*, Aug. 2014, pp. 153–162.

[48] S. Ali, L. Hongqi, S. U. Khan, Y. Zhongguo, and Z. Liping, "Success factors for software outsourcing partnership management: An exploratory study using systematic literature review," *IEEE Access*, vol. 5, pp. 23589–23612, 2017.

[49] E. Gamma, *Design Patterns: Elements of Reusable Object-Oriented Software*. London, U.K.: Pearson, 1995.

[50] S. Robertson, "Requirements patterns via events/use cases," Proc. Pattern Lang. Program. (PLoP), Washington Univ., Tech. Rep. #wucs-97-07, 1996.

[51] M. Fowler, *Analysis Patterns: Reusable Object Models*. Reading, MA, USA: Addison-Wesley, 1997.

[52] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *A System of Patterns–Pattern Oriented Software Architecture*. Hoboken, NJ, USA: Wiley, 1996.

[53] W. J. Brown, R. C. Malveau, H. W. S. McCormick, and T. J. Mowbray, *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*. Hoboken, NJ, USA: Wiley, 1998.

[54] M. Kausar and A. Al-Yasiri, "Distributed agile patterns for offshore software development," in *Proc. 12th Int. Joint Conf. Comput. Sci. Softw. Eng. (JCSSE)*, 2015, pp. 1–6.

[55] J. Noll, I. Richardson, and S. Beecham, "Patternizing GSD research: Maintainable decision support for global software development," in *Proc. IEEE 9th Int. Conf. Global Softw. Eng.*, Aug. 2014, pp. 110–115.

[56] C. Lescher, "Patterns for global development: how to build one global team?" in *Proc. 15th Eur. Conf. Pattern Lang. Programs*, 2010, pp. 1–6.

[57] U. van Heesch, "Collaboration patterns for offshore software development," in *Proc. 20th Eur. Conf. Pattern Lang. Programs*, Jul. 2015, pp. 1–10.

[58] L. Cordeiro, C. Becker, and R. Barreto, "Applying scrum and organizational patterns to multi-site software development," in *Proc. 6th Latin Amer. Conf. Pattern Lang. Program.*, Porto de Galinhas, Brazil, 2007, pp. 46–67.

[59] A. Välimäki and J. Kääriäinen, "Patterns for distributed scrum—A case study," in *Enterprise Interoperability III*. London, U.K.: Springer, 2008, pp. 85–97.

[60] M. Paasivaara and C. Lassenius, "Communities of practice in a large distributed agile software development organization–case Ericsson," *Inf. Softw. Technol.*, vol. 56, no. 12, pp. 1556–1577, 2014.

[61] G. Papadopoulos, "Moving from traditional to agile software development methodologies also on large, distributed projects," *Proc.-Social Behav. Sci.*, vol. 175, pp. 455–463, Feb. 2015.

[62] S. Hole and N. B. Moe, "A case study of coordination in distributed agile software development," in *Proc. Eur. Conf. Softw. Process Improvement*. Springer, 2008, pp. 189–200.

[63] A. Putta, "Scaling agile software development to large and globally distributed large-scale organizations," in *Proc. 13th Int. Conf. Global Softw. Eng.*, May 2018, pp. 141–144.

[64] M. Paasivaara and C. Lassenius, "Collaboration practices in global inter-organizational software development projects," *Softw. Process: Improvement Pract.*, vol. 8, no. 4, pp. 183–199, Oct. 2003.

[65] L. B. Hvatum, V. Bricout, D. Heliot, A. Cretoiu, Y. Yang, and T. Simien, "Patterns for managing distributed product development teams," in *Proc. 9th Eur. Conf. Pattern Lang. Programs (EuroPLoP)*, 2004, pp. 109–122.

[66] A. Välimäki, J. Kääriäinen, and K. Koskimies, "Global software development patterns for project management," in *Proc. Eur. Conf. Softw. Process Improvement*. Springer, 2009, pp. 137–148.

[67] C. Hofmann, S. Lauber, B. Haefner, and G. Lanza, "Development of an agile development method based on Kanban for distributed part-time teams and an introduction framework," *Proc. Manuf.*, vol. 23, pp. 45–50, Jan. 2018.

[68] R. A. Khan, M. Y. Idris, S. U. Khan, M. Ilyas, S. Ali, A. U. Din, G. Murtaza, and A. W. Wahid, "An evaluation framework for communication and coordination processes in offshore software development outsourcing relationship: Using fuzzy methods," *IEEE Access*, vol. 7, pp. 112879–112906, 2019.

[69] A. K. Aggarwal and V. S. Mani, "Using product line engineering in a globally distributed agile development team to shorten release cycles effectively," in *Proc. ACM/IEEE 14th Int. Conf. Global Softw. Eng. (ICGSE)*, May 2019, pp. 58–61.

[70] R. A. Mahajan and S. A. Mahajan, "Development of scrum-tree-KNN algorithm for distributed agile development," in *Proc. Int. Conf. Emerg. Smart Comput. Informat. (ESCI)*, Mar. 2020, pp. 17–21.

[71] S. A. Licorish and S. G. MacDonell, "How do globally distributed agile teams self-organise? Initial insights from a case study," in *Proc. 8th Int. Conf. Eval. Novel Approaches Softw. Eng. (ENASE)*, 2013, pp. 157–164.

[72] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele Univ., U.K., EBSE Tech. Rep. EBSE-2007-01, 2007.

[73] K. Krippendorff, "Reliability in content analysis: Some common misconceptions and recommendations," *Human Commun. Res.*, vol. 30, no. 3, pp. 411–433, Jul. 2004.

[74] N. Kerth, *Project Retrospectives: A Handbook for Team Reviews*. Reading, MA, USA: Addison-Wesley, 2001.

[75] E. Hossain, M. A. Babar, and H.-Y. Paik, "Using scrum in global software development: A systematic literature review," in *Proc. 4th IEEE Int. Conf. Global Softw. Eng.*, Jul. 2009, pp. 175–184.

[76] M. Cottmeyer, "The good and bad of agile offshore development," in *Proc. Agile Conf.*, Aug. 2008, pp. 362–367.

[77] A. Avritzer, F. Bronsard, and G. Matos, "Improving global development using agile," in *Agility Across Time and Space, Implementing Agile Methods in Global Software Projects*, D. Smite, N. B. Moe, and P. J. Ägerfalk, Eds. Berlin, Germany: Springer, 2010, pp. 133–148.

[78] M. Paasivaara, S. Durasiewicz, and C. Lassenius, "Using scrum in distributed agile development: A multiple case study," in *Proc. 4th IEEE Int. Conf. Global Softw. Eng.*, Jul. 2009, pp. 195–204.

[79] K. Sureshchandra and J. Shrinivasavadhani, "Adopting agile in distributed development," in *Proc. IEEE Int. Conf. Global Softw. Eng.*, Aug. 2008, pp. 217–221.

[80] L. M. Maruping, "Implementing extreme programming in distributed software project teams: Strategies and challenges," in *Agility Across Time and Space*. Berlin, Germany: Springer, 2010, pp. 11–30.

[81] M. Vax and S. Michaud, "Distributed agile: Growing a practice together," in *Proc. Agile Conf.*, Aug. 2008, pp. 310–314.

[82] S. Berczuk, "Back to basics: The role of agile principles in success with an distributed scrum team," in *Proc. Conf. (AGILE)*, J. Eckstein, F. Maurer, R. Davies, G. Melnik, and G. Pollice, Eds. Washington, DC, USA: IEEE Computer Society, Aug. 2007, pp. 382–388.

[83] M. Cristal, D. Wildt, and R. Prikladnicki, "Usage of SCRUM practices within a global company," in *Proc. IEEE Int. Conf. Global Softw. Eng.*, Aug. 2008, pp. 222–226.

[84] A. Danait, "Agile offshore techniques–A case study," in *Proc. Agile Develop. Conf. (ADC)*, Jul. 2005, pp. 214–217.

[85] B. Ramesh, L. Cao, K. Mohan, and P. Xu, "Can distributed software development be agile?" *Commun. ACM*, vol. 49, no. 10, pp. 41–46, 2006.

[86] P. G. Armour, "Agile…and offshore," *Commun. ACM*, vol. 50, no. 1, pp. 13–16, 2007.

[87] J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, "Distributed scrum: Agile project management with outsourced development teams," in *Proc. 40th Annu. Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2007, p. 274.

[88] M. Yap, "Follow the sun: Distributed extreme programming development," in *Proc. Agile Develop. Conf. (ADC)*, Jul. 2005, pp. 218–224.

[89] C. Kussmaul, R. Jack, and B. Sponsler, "Outsourcing and offshoring with agility: A case study," in *Proc. Conf. Extreme Program. Agile Methods*. Berlin, Germany: Springer, 2004, pp. 147–154.

[90] I. Bose, "Lessons learned from distributed agile software projects: A case-based analysis," *Commun. Assoc. Inf. Syst.*, vol. 23, no. 1, p. 34, 2008.

[91] S. Modi, P. Abbott, and S. Counsell, "Negotiating common ground in distributed agile development: A case study perspective," in *Proc. IEEE 8th Int. Conf. Global Softw. Eng.*, Aug. 2013, pp. 80–89.

[92] M. Korkala, M. Pikkarainen, and K. Conboy, "Combining agile and traditional: Customer communication in distributed environment," in *Agility Across Time and Space*. Berlin, Germany: Springer, 2010, pp. 201–216.

[93] M. Summers, "Insights into an agile adventure with offshore partners," in *Proc. Agile Conf.*, Aug. 2008, pp. 333–338.

[94] D. Wildt and R. Prikladnicki, "Transitioning from distributed and traditional to distributed and agile: An experience report," in *Agility Across Time and Space*. Berlin, Germany: Springer, 2010, pp. 31–46.

[95] B. S. Drummond and J. F. Unson, "Yahoo! Distributed agile: Notes from the world over," in *Proc. Agile Conf.*, 2008, pp. 315–321.

[96] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen, "Large-scale agile transformation at ericsson: A case study," *Empirical Softw. Eng.*, vol. 23, no. 5, pp. 2550–2596, Oct. 2018.

[97] C. J. Poole, "Distributed product development using extreme programming," in *Proc. Int. Conf. Extreme Program. Agile Processes Softw. Eng.* Berlin, Germany: Springer, 2004, pp. 60–67.

[98] A. W. Brown, "A case study in agile-at-scale delivery," in *Proc. Int. Conf. Agile Softw. Develop.* Berlin, Germany: Springer, 2011, pp. 266–281.

[99] S. Ali, H. Li, S. U. Khan, Y. Zhao, and L. Li, "Fuzzy multi attribute assessment model for software outsourcing partnership formation," *IEEE Access*, vol. 6, pp. 55431–55461, 2018.

[100] S. Ali, J. Huang, S. U. Khan, and H. Li, "A framework for modelling structural association amongst barriers to software outsourcing partnership formation: An interpretive structural modelling approach," *J. Softw., Evol. Process*, vol. 32, no. 6, Jun. 2020, Art. no. e02243.

[101] M. Niazi, D. Wilson, D. Zowghi, and B. Wong, "A model for the implementation of software process improvement: An empirical study," in *Product Focused Software Process Improvement*, F. Bomarius and H. Iida, Eds. Berlin, Germany: Springer 2004, pp. 1–16.

MARYAM KAUSAR received the B.S. degree in computer science from the NUCES-FAST, Pakistan, in 2010, the M.S. degree in computer science and IT management from The University of Manchester, U.K., in 2011, and the Ph.D. degree from the School of CSE, University of Salford, U.K., in 2018. Since 2019, she has been working as an Assistant Professor at Foundation University Islamabad, Pakistan. Her research interests include software engineering, requirement engineering, global software development, and agile software development. Her awards and honors include securing funding for developing an augment reality game using agile methods and being the Chair of the Pakistan Agile Development Society.

M. ISHTIAQ received the B.S. degree from the Virtual University of Pakistan, in 2008, and the M.S. and Ph.D. degrees from the National University of Computer and Emerging Sciences, Islamabad, in 2009 and 2018, respectively. He worked in different positions at the National University of Computer and Emerging Sciences and at the COMSATS Institute of Information Technology. He joined Foundation University Islamabad, as an Assistant Professor, in 2014. His research interests include machine learning, and multimedia security and forensics.

SHARIQ HUSSAIN received the M.Sc. degree from PMAS Arid Agriculture University, Rawalpindi, Pakistan, and the Ph.D. degree from the University of Science and Technology Beijing. Since 2014, he has been an Assistant Professor with Foundation University Islamabad. His research interests include software engineering, web services, and the IoT.