Tech Science Press

Framework for Effective Utilization of Distributed Scrum in Software Projects

Basit Shahzad[1], Wardah Naeem Awan[1,*], Fazal-e-Amin[2], Ahsanullah Abro[3], Muhammad Shoaib[4] and Sultan Alyahya[4]

[1]Department of Software Engineering, National University of Modern Languages, Islamabad, 44000, Pakistan
[2]Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, 11543, Saudi Arabia
[3]Department of Computer Science, Sukkur IBA University, Sindh, 65200, Pakistan
[4]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh, 11543, Saudi Arabia
*Corresponding Author: Wardah Naeem Awan. Email: wardah.naeem.awan@gmail.com

Abstract: There is an emerging interest in using agile methodologies in Global Software Development (GSD) to get the mutual benefits of both methods. Scrum is currently admired by many development teams as an agile most known methodology and considered adequate for collocated teams. At the same time, stakeholders in GSD are dispersed by geographical, temporal, and socio-cultural distances. Due to the controversial nature of Scrum and GSD, many significant challenges arise that might restrict the use of Scrum in GSD. We conducted a Systematic Literature Review (SLR) by following Kitchenham guidelines to identify the challenges that limit the use of Scrum in GSD and to explore the mitigation strategies adopted by practitioners to resolve the challenges. To validate our review findings, we conducted an industrial survey of 305 practitioners. The results of our study are consolidated into a research framework. The framework represents current best practices and recommendations to mitigate the identified distributed scrum challenges and is validated by five experts of distributed Scrum. Results of the expert review were found supportive, reflecting that the framework will help the stakeholders deliver sustainable products by effectively mitigating the identified challenges.

1 Introduction

Global software development (GSD) is a recent ongoing trend in the software development industry [1], that usually engage stakeholders situated in various geographic locations and time zones, from different national and organizational cultures. It provides approach to the organization to competent and skillful resources at a lesser cost and easy accessibility to the markets [2]. GSD also has some challenges, which is not part of (or not as prominent) in traditional co-located development i.e., geographical, temporal, and

socio-cultural distances that can lead to meaningful communication, coordination, and collaboration challenges that must be controlled to achieve the benefits of GSD [3].

There is a rising interest of implementing agile practices in GSD projects to get the mutual benefits of both methods [1]. Scrum, as an Agile common and most important practice, recently admired by many software teams in Agile distributed development. According to survey the scrum method of Agile software development is used over by 89% of agile development teams [3]. From the latest survey about agile practice acceptance rate, it is stated that agile methods could be effectively used by considerably dispersed members of a team. Another survey stated that, from the several agile practices, Scrum practices have a greater acceptance rate. Hence, it can be argued that the Scrum, which is the main component of agile is becoming popular gradually and could be effectively used by the teams that are dispersed globally [4].

As scrum teams are self-organized, enabled on excellent team collaboration and communication. Hence, it is hard to directly implement Scrum practices in GSD because in GSD development, team members are distributed physically by time-based, geographical, and socio-cultural distances resulting in numerous challenges or issues that might affect teams' communication and collaboration process [1]. There are multiple challenges confronted by man organizations who are using scrum practices in GSD [5].

We conducted up-to-date SLR to identify the challenges confronted by distributed scrum teams and the mitigation strategies adopted by teams to overcome the confronted challenges. While conducting the pilot review, it was observed that some studies have already proposed frameworks to mitigate the challenges. Still, their proposed frameworks are for resolving some specific challenges i.e., in [2], a framework is proposed to mitigate communication challenges. Similarly, in [3] author proposed a framework to manage risks in distributed Scrum, and most of the frameworks identified in the literature was unevaluated. To our best knowledge, we did not find any study that has developed a framework by identifying the latest challenges of distributed Scrum with the framework to mitigate all the confronted challenges.

Therefore, we aimed to identify and explore all the challenges by conducting up-to-date SLR and develop a framework to mitigate the identified challenges. The findings of this study will help the practitioners to understand the challenges that could be confronted while doing development in distributed scrum, and our proposed framework will help them to mitigate the challenges effectively to ensure sustainable development.

The rest of the paper is organized in the following different sections: Section 2 discusses the background and related work. Section 3 describes the research method. Section 4 describes the research questions. Section 5 discusses the proposed framework, while Section 6 presents the results and discussion. Section 7 focuses on validity issues. Finally, Section 8 concludes with the implications of the study and guidance for future research.

2 Background

This section discusses the overview of the scrum framework and related work of our study.

2.1 Overview of Scrum Process

Scrum is a hybrid model which is the combination of iterative and incremental models [3]. Using Scrum, the software is delivered in different increments called "Sprints" (typically 2–4 weeks iterations). Every sprint starts with the planning of sprint and ends with sprint review. Sprint planning is a time-bound meeting from the scrum team and could be last for up to 4 hours. The meeting is devoted to the development of detailed plans for the sprint. After that sprint review meeting is conducted, that is attended by the stakeholders of the projects to assess the business conditions, the market, and the technology. The

review meetings possibly will also be last for more than 4 hours. A retrospective meeting could also be planned to evaluate the teamwork in finalized sprints. Day-to-day scrum meeting conducted by the team is 15-minute long in which every participant of the team addresses the questions what I did yesterday, what is my today's task, and what obstacles are there to perform my task? Scrum creates 3 artifacts: product backlogs, sprint backlogs, and burn-down charts. Backlogs contain requirements from customers, and daily burn down indicates the total remaining work [4]. Fig. 1 shows the overview of scrum process.
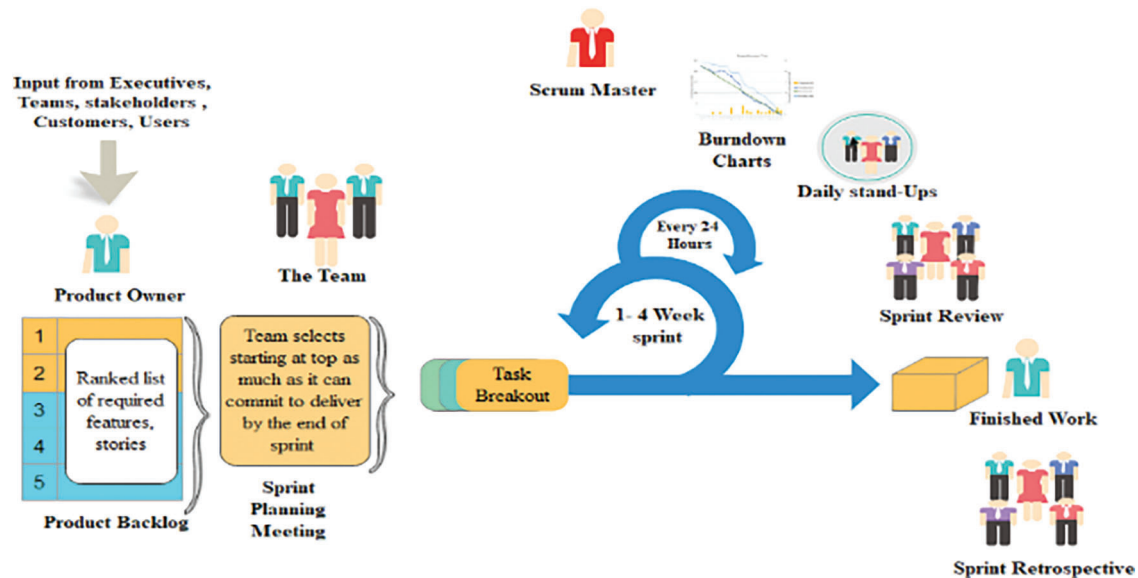
Figure 1: Overview of Scrum Process

2.2 Distributed Scrum Model

There are three commonly observed and used model of distributed Scrum i.e., Isolated Scrum, Distributed Scrum of Scrum and Totally Integrated Scrum [4].

2.2.1 Isolated Scrum

In this model, scrum teams are isolated geographically. Mostly offshore teams are either not cross-functional or not using the process of Scrum [4].

2.2.2 Distributed Scrum of Scrum

In this model, scrum teams are distributed geographically but are connected through regular SoS meetings [4].

2.2.3 Totally Integrated Scrum

In this model, teams are geographically distributed and are cross-functional with the members that are distributed among geographies [4].

2.3 Related Work

Agile approaches have been originally designed for co-located software development, and it is hard to directly apply these methods to distributed development [6] and have a lots of challenges and issues while executing scrum in distributed and GSD, that needs to be discussed to overcome the challenges and issues from distributed Scrum to ensure sustainable development.

This paper [7] discussed the patterns and the challenges related to the development patterns used in distributed scrum development. Different challenges were identified, and a case study was performed. The outcomes of the case study show that distributed scrum has many benefits if it gets implemented successfully, and the members of the team are more satisfied with the newly distributed scrum as compared to traditional approaches, even though there are still some challenges that must be addressed and resolved in upcoming improvement efforts. The limitation of the paper is that the analysis of proposed patterns is not performed.

Paper [8] performed a case study on the organization which is using scrum practices. 22 semi-structured interviews were conducted to find out the challenges of scaling Scrum. Different challenges are confronted while implementing agile practices such as Scrum-of-Scrums, area product owners, common retrospective, common sprint demo, and common sprint planning. It is found that practical implementation of scaling distributed scrum is not as easy as it seems in theoretical manners in books. The limitation of the study is that most of the interviews were conducted from the main site so, there are chances of bias as results might be different if other sites have been involved.

In [9] authors conducted SLR to determine the human-related factors that can negatively influence while adopting agile in GSD. 11 different challenges were found, and 6 out of 11 challenges found critical challenges having the frequency of ≥50%. The challenges that were found critical were: lack of communication with the highest frequency of (88%), lack of customer involvement having a frequency (83%), lack of management commitments with the frequency of (72%), lack of requirements analysis with the frequency of (56%), lack of knowledge sharing (56%) and lack of roles and responsibilities (50%). A hypothetical model was developed based on identified challenges, which shows the relationship between identified challenges and scaling agile in GSD.

This paper [10] discuss an explanatory case study conducted in Norway for four years, on one of the largest software development programs to find out how agile methods are evolved to use for large scale and distributed development and what are the challenges encountered while using agile method in large scale development. It is found that communication, coordination and collaboration, customer involvement, knowledge sharing, and architecture management are the most challenging areas while using agile methods in large-scale development. In paper [11], SLR is conducted to determine the themes and primary issues of distributed agile development. It is found that project planning and management, communication, coordination, collaboration, cultural and quality assurance issues are the major concerns that need to be addressed and resolved to execute agile methodology in a distributive manner effectively. This paper limits in a way that it includes only articles published in IEEE, so the findings of this paper are limited in scope.

3 Research Methodology

This paper explained a framework proposed by consolidating best practices, adopted by different teams to mitigate distributed scrum challenges. We conducted SLR by reviewing articles from 2005 to 2021 by following Kitchenham [12] to identify all the challenges confronted by distributed scrum teams and mitigation strategies adopted by teams to overcome the challenges. Out of 3348 articles, we selected 56 articles by applying different filters that discussed distributed scrum challenges and mitigation strategies adopted by teams to overcome the challenges. Twelve significant challenges were identified through SLR, shown in Tab. 1.

Table 1: List of identified challenges

No	Challenges	References	Frequency N = 56	Percentage (%)
1	Communication, Coordination & Collaboration	[2, 4, 5, 7–54]	51	91%
2	Knowledge Sharing and Management	[6, 10, 15, 18, 22, 23, 28–30, 38, 39, 50, 51, 53–55]	16	28%
3	Project Management	[6, 7, 9, 21, 22, 34, 39, 46, 48, 50, 52, 54, 56]	13	25%
4	Technical Issues	[6, 7, 9–11, 18, 22, 29, 30, 34, 36, 42, 47, 50, 55]	15	28%
5	Trust Issues	[9, 10, 13, 23, 25, 29, 36, 43, 47, 48, 54, 34]	12	22%
6	Requirement Engineering	[6, 18, 21, 24, 39, 44–46, 48–50]	11	19%
7	Quality Assurance	[6, 28, 35, 44, 49, 50, 52]	7	12%
8	Configuration/Integration Management Issues	[6, 10, 15, 18, 28, 44, 45, 50]	8	14%
9	Resistance to change	[15, 35, 44, 45]	4	7%
10	Risk Management	[10, 33]	2	3%
11	Architectural Issues	[6, 10, 34, 46, 48, 49, 51]	7	12%
12	Shared understanding	[9, 16, 29, 54, 57]	5	9%

After identifying challenges, we conducted an industrial survey from 305 practitioners by following Kasunic guidelines [13] to validate the challenges identified through SLR to accept or reject the challenges. We used a 5-point Likert scale to ask questions, and two types of Likert scales i.e., agreement and frequency are used for that. The values used for both of Likert scales were ranging from "–2 to 2" and "0 to 4".

To evaluate our scales' internal consistency and reliability to conduct a survey, we used Cronbach alpha's value. As we have used 2 different Likert scales so, the first calculated value of Cronbach alpha was "0.85" and covered seven factors from the total 12 factors. Another type of likert scale covered the remaining five factors, and the calculated value for this scale against five factors was "0.86".

We set a criterion to accept and rejected each challenge and the challenges with an average value greater or equal to 0.90 for 1st type of likert scale got accepted, and from 2nd likert scale, factors with an average value greater or equal to 1.50 got accepted and remaining factors got rejected. Out of 12 factors, 8 factors got accepted by practitioners. Tab. 2 shows the challenges accepted and rejected through survey, and Fig. 2 shows the average weightage response against each factor.

Next, a research framework is proposed by consolidating the best practice identified through SLR to mitigate the validated challenges through a survey. We conducted an expert review by interviewing 5 experts of distributed scrum, to evaluate the effectiveness of the proposed framework and ensure that the proposed framework will be helpful in sustainable development.

Table 2: Accepted and rejected challenges

No	Factors	Weightage values	Avg. weightage responses	Results
F1	Software Architectural Understanding	713	2.33	Accepted
F2	Requirement Engineering	695	2.27	Accepted
F3	Shared Understanding	681	2.23	Accepted
F4	Knowledge Sharing and Management	673	2.20	Accepted
F5	Project Management	647	2.12	Accepted
F6	Trust Issues	579	1.89	Accepted
F7	Communication, Coordination & Collaboration	347	1.13	Accepted
F8	Technical Issues	337	1.10	Accepted
F9	Risk Management	229	0.75	Rejected
F10	Lack of resistance to Change	228	0.74	Rejected
F11	Quality Assurance	196	0.64	Rejected
F12	Configuration/Integration Management	168	0.55	Rejected
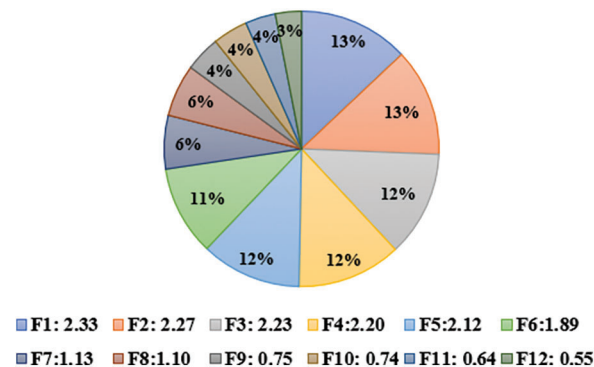
Figure 2: Average weightage response against each factor

4 Research Questions

While conducting pilot studies, we noticed that some studies have already proposed frameworks to mitigate the challenges. Still, their proposed frameworks are for the resolution of some specific challenges. To our best knowledge, we did not find any study that has identified the latest challenges of distributed Scrum and proposed a single framework to mitigate all the identified challenges.

To address this gap, the following research questions were formulated:

RQ1: What are the challenges confronted by teams during the execution of Scrum in a distributive manner?

RQ2: What are the mitigation strategies adopted by teams to treat the confronted challenges?

Fig. 2 shows the challenges accepted and rejected by practitioners through surveys along with average weightage response against each factor.

5 Proposed Framework

We developed a conceptual framework based on the identified challenges and mitigation strategies adopted by teams in distributed Scrum. The elements of our framework with its possible usage are discussed below.

5.1 Framework Development Process

To develop a conceptual framework, we carefully followed the few planned steps mentioned below:

- To identify the components of our conceptual framework, we reviewed the several frameworks discussed in our selected studies i.e., ([9,11,14,15,16]).
- We conducted SLR to identify the challenges confronted by distributed scrum teams and then survey was conducted from practitioners to validate those identified challenges.
- Reviewed selected articles to identify the mitigation strategies adopted by distributed scrum teams to resolve or mitigate the confronted challenges.
- Finally, we consolidated the validated challenges by practitioners and best practices adopted by different teams to resolve those challenges, into a framework.

5.2 Components of Framework

Our framework is broadly composed of 2 main components:

   i) Major Challenges
  ii) Mitigation Strategies

Fig. 3 shows the framework we proposed. We have discussed each challenge along the mitigation strategies to reduce or overcome the challenges.
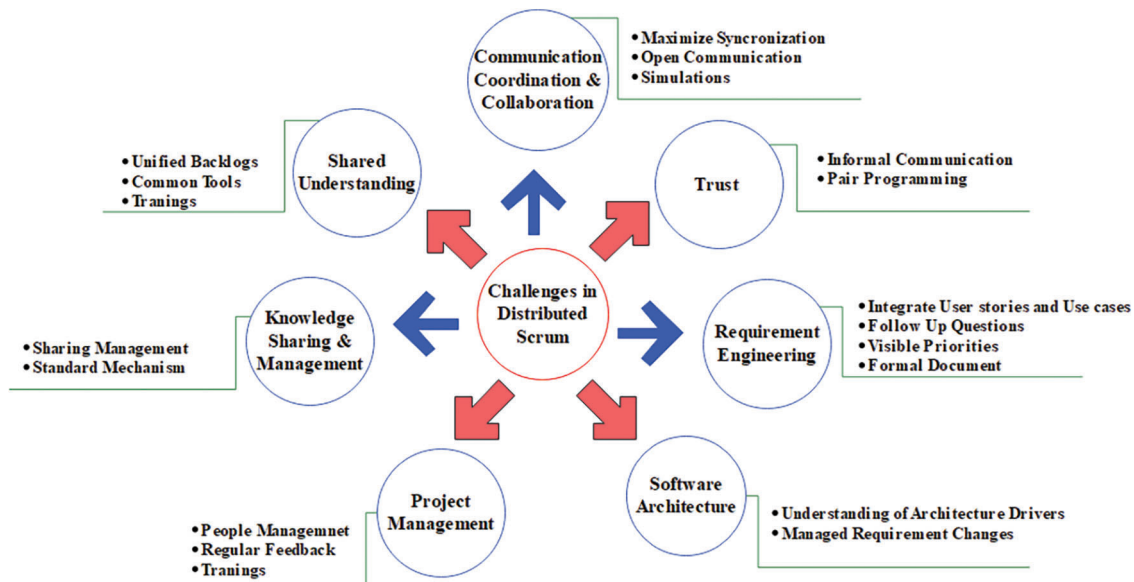

Figure 3: Proposed framework

5.3 Communication, Coordination and Collaboration

Scrum is enabled on great communication, coordination, and collaboration among team so, it is the biggest challenge identified through literature and survey. Coordination and collaboration among teams depend upon communication so, communication is the key to better coordination and collaboration among teams [17]. Different resolution strategies are being by different teams to overcome these issues. We have identified some best strategies to overcome these issues.

5.3.1 Maximize Synchronized Working Hours

This is a widely used strategy in distributed scrum development to improve effective communication among teams. As teams are working from different locations, with varying hours of working due to temporal differences, so project owners try to adjust overlapping working hours of distributed teams by allowing team members to attend meetings from home [1,9,16–25].

5.3.2 Open Communication

Open communication allow the team members to openly discuss their views about things and allow them to add suggestions in things, whenever required, that leads to better coordination and collaboration among teams [9,21,26–28].

5.3.3 Simulations

As Scrum require face-to-face communication, which is not possible in distributed environment, so try to replace direct face-to-face communication with rich communication channels through simulations. Using high quality simulation tools for face-to-face discussion will help the teams to communicate effectively, leads to better coordination and collaboration among distributed teams [5,23,24,29].

5.4 Trust

Trust is another significant challenge reported in the literature. Different factors are identified from the literature, including ineffective communication, coordination, and collaboration among teams, poor team bonding, cultural barriers, etc., that causes trust issues in distributed Scrum. Some best strategies we have identified from the literature include:

5.4.1 Informal Communication

Informal communication is the strategy adopted by different teams, to build trust among team members. By adopting different modes of informal communication (i.e., individual or teleconferences, video conferences, emails, instant messages) with formal documentation can help to build trust among team members [14,19,25–30].

5.4.2 Pair Programming

It is another strategy adopted by teams to build trust among distributed teams by encouraging pair programming among teams. When team members collaborate to share things and experiences, it will positively affect the overall performance of teams [21,31–33].

5.5 Requirement Engineering

Although the concept of requirement engineering and scrum method seems incompatible because requirement engineering is all about heavy documentation throughout requirement engineering process and scrum does not demand huge documentation but, the team who are working in distributed Scrum are facing different challenges associated with requirement engineering due to many reasons including lack of formal documentation, invisible priorities of requirements, ambiguous and unclear requirements. The best resolution strategies we have identified from literature to resolve the requirement engineering issues include:

5.5.1 Integrate User Stories with Use Cases

Unsynchronized user stories and use cases lead to misunderstandings among development teams about work progress and completion of work, primarily when teams work in a distributed environment. To avoid misunderstanding about work progress, it is important to integrate user stories with use cases before starting actual development [21,25,27,34–37].

5.5.2 Follow-Up Questions

Follow-Up questions are very important to ensure that all the stakeholders have same understanding about the functionality of requirements. As the chances of misunderstood requirements are higher in distributed development. Correct and clear understanding of requirements can be ensured by asking follow-up question [13,28,38–42].

5.5.3 Visible Requirement Priorities

Continuously changing priority of requirements lead to changes in software architecture, result in software architectural understanding issues. To resolve these issues, practitioners suggested finalizing the requirement and prioritizing them clearly before starting the actual development [9,20,41,43–45].

5.5.4 Formal Document

Although scrum do not require heavy documentation but, lack of documentation in distributed scrum development leads to many challenges, i.e., misunderstood or unclear requirements, requirement traceability issues, requirement change management, etc. To avoid these issues, it is important to maintain a formal requirement document with a standard template understandable to all distributed teams [19,27–30,46–49].

5.6 Software Architectural Understanding

A clear understanding of software architecture is essential for the success of any project. Usually, the architecture of any software is based on 3–4 critical requirements [30]. As agile methods welcome changes at any stage of development so, teams working in distributed environments find it challenging because sometimes product owner could not deliver understandable requirements to the teams due to continuous changes in requirements and with the constant change in requirements, change in requirements remain unmanaged that lead to software architectural understanding issues. To mitigate this challenge, different teams adopt different practices. The best practice we identified for our framework includes:

5.6.1 Understanding of Architecture Drivers

To equally understand the critical requirements, devote multiple iterations to finalize those critical requirements on which the software architecture is based upon, and design high level architecture, to avoid the changes in critical requirements and the conflict about critical requirements among teams [50–53].

5.6.2 Managed Requirement Changes

Maintain a unified repository with the traceability process for requirement changes, accessible to all development teams and teams get notified whenever any change is requested against any predefined requirements. It helps to timely exchange the information between product owner and development teams, regarding the request of change in requirement by customers and the possibility about the acceptance of those change requests with their effect on overall development process [29,30,33–36].

5.7 Project Management

Project management itself complex task and managing a project in distributed environment is an even more difficult task due to geographical, socio-cultural distance, and temporal differences. Product owners adopt different strategies to mitigate project management issues. Some best strategies include:

5.7.1 People Management

While assigning the task to the teams, clearly consider the competencies required to accomplish that task, and after assigning the task to the teams, clearly define the responsibilities of each member of team related to a specific task, and use tools to track the progress of the team and the work [8,23,54–59].

5.7.2 Regular Feedback

Get regular feedback from stakeholders about work by regularly communicating with them to ensure that the team is progressing in the right direction [12,44,60–64].

5.7.3 Training

Providing training or coaching to teams about work whenever is required or demanded by teams, to teach common skills to all distributed teams. It help the teams timely deliver the exact product demanded by customers. Allowing or sponsoring the teams to attend the training organized by another organization help the teams to enhance their skills in latest technology results in better performance from teams [35,65–71].

5.8 Knowledge Sharing and Management

To mitigate the knowledge sharing and management challenge in distributed scrum development, different strategies are defined in the literature. The best practices that we have identified are:

5.8.1 Sharing Management

By maintaining a unified backlog that gets updated timely according to work progress and by maintaining a knowledge management system that can track the performance and progress of the teams about the work to be done and completed work and access to all the teams can help the teams to resolve this knowledge sharing and management issues as common data will be visible to all teams [40,72–76].

5.8.2 Standard Mechanism

By defining a proper mechanism and standard template to share data (i.e., standard template for project documentation for all the teams) among distributed teams can help to cover the knowledge sharing and management issues [50–54,73,77–79].

5.9 Shared Understanding

To ensure the equal understanding of the team members about the work to be done, different strategies are adopted by product owners as lack to shared understanding among teams can lead to project failure. Some best practices identified from literature to ensure shared understanding includes:

5.9.1 Trainings

By providing training about different cultures in different languages can help to encourage the teams to communicate openly without hesitation and can help to overcome language barrier [16,80–83].

5.9.2 Common Tools

By ensuring the usage of common tools for development by all the distributed teams working on similar projects can help the teams to easily understand the data or information shared with them [9,59–62].

5.9.3 Unified Backlog

By maintaining a unified backlog having clearly visible requirements for all the teams can help to overcome the issue of shared understanding, as all teams will be able to access and view the common data along with the progress of the work in terms of completed sprints and current sprints [22,32,63,78,84–86].

5.10 Technical Issues

To ensure sustainable development, the teams need to be technically advanced, especially when teams are working in distributed Scrum as, Scrum is enabled on continuous communication and collaboration among teams. Effective and continuous communication and collaboration among distributed teams is possible only when teams are equipped with advanced technology. From literature, it is reported that many distributed scrum teams are facing technical dependency challenges due to lack of effective tools support for scrum processes, ineffective communication tools, lack of trainings, lack of globally shared unified backlogs and lack of information about organizational infrastructures. It is important to mitigate all the technical dependency issues to develop and deploy sustainable product while working in distributed scrum [9,86]. Technical issues can be mitigated by focusing on all other challenges identified in this study.

6 Results and Discussion

We conducted SLR to identify the challenges confronted by distributed scrum teams while developing and the mitigation strategies adopted by teams to resolve the challenges. From the selected 56 studies, 12 major challenges were identified. Tab. 1 shows the challenges identified through SLR along with their frequency of occurrence, i.e., 91% of studies chosen discusses communication, coordination, and collaboration issues, 28% of studies discusses knowledge sharing and management issues, 28% of studies discusses technical issues, and 25% of studies discusses project management issues.

After identifying challenges, we conducted an industrial survey to validate the challenges identified through SLR by practitioners to accept or reject the challenges. Out of 12 challenges, eight challenges were accepted by following defined criteria based on average weightage response against each challenge. The remaining challenges were rejected. Tab. 2 shows the accepted and rejected challenges with average weight against each challenging factor and Fig. 2 shows the average weightage response against each factor.

We proposed a framework to mitigate the accepted challenges by consolidating the best practices identified through SLR and conducted an expert review by conducting interviews of 5 distributed scrum experts to identify the effectiveness of proposed framework. Results of expert review revealed that proposed framework will effectively contribute to sustainable development by using distributed Scrum. Fig. 3 shows our proposed framework.

7 Threats to Validity

Although we conducted this study by following well-defined and thorough research instruments [12,13], following threats might exist to this study:

1. Articles selected for this study were searched by single researcher only, so there is a chance that we could have missed some important studies. Though, we used multiple databases and snowball sampling techniques, which reduces the possibilities of missing relevant studies.
2. We have not conducted any industry-based case study to evaluate our proposed framework.

8 Conclusion and Future Work

There is an emerging concern of using agile practices in GSD projects to get the mutual benefits of both distributive and agile methods. Scrum, as an agile most known methodology, is currently admired by many development teams. It is typically considered productive for small-scale projects with co-located teams because scrum teams are self-organized and enabled on excellent team collaboration and communication. However, GSD are usually distributed by distance differences, which results in numerous challenges or risks that might affect team collaboration and communication processes. For sustainable development and

implementation of distributed Scrum in GSD, there is a need to identify and mitigate all the confronted challenges in distributed Scrum. This study identifies challenges associated with distributed scrum development by analyzing past studies and validated by practitioners through a survey. A framework is proposed to mitigate the challenges validated by practitioners, and expert review is conducted by interviewing five distributed scrum experts to validate the proposed framework. The focus of this study was the identification of distributed scrum challenges and the proposal of a framework to mitigate the identified challenges for sustainable distributed development.

An extensive systematic literature review was conducted to ensure the maximum validity of the study. Selection criteria was defined to extract the studies, and 3348 studies were extracted initially from different renowned databases. After careful review of the studies, 56 articles were selected in response to RQ1 and RQ2. By performing a critical review on the studies, 12 major challenges were identified, and out of 12 challenges, distributed scrum practitioners accepted eight challenges through the survey. This study revealed that lack of communication, coordination, and collaboration among distributed scrum teams is the most critical challenge with frequency of 91%, and it leads to many other challenges.

Efforts have been made to cover the maximum number of distributed scrum challenges and mitigation strategies available in the literature to propose a framework. Our proposed framework is expected to help distributed scrum practitioners to mitigate the confronted challenges while working in distributed Scrum effectively.

We plan to conduct an industrial case study by using the proposed framework to evaluate the further effectiveness of the framework and ensure sustainable distributed scrum development.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] E. Hossain, P. L. Bannerman and D. R. Jeffery, "Scrum practices in global software development: A research framework," *Journal of Software Engineering Research and Development*, vol. 6759, no. 3, pp. 88–102, 2011.

[2] B. Shahzad, A. M. Abdullatif, N. Ikram and A. Mashkoor, "Build software or buy: A study on developing large scale software," *IEEE Access*, vol. 5, no. 3, pp. 24262–24274, 2017.

[3] A. Al-Zaidi and R. Qureshi, "Global software development geographical distance communication challenges," *International Arab Journal of Information Technology*, vol. 14, no. 2, pp. 215–222, 2017.

[4] M. Esteki, T. J. Gandomani and H. K. Farsani, "A risk management framework for distributed scrum using prince2 methodology," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1299–1310, 2020.

[5] K. Saleem, B. Shahzad, M. A. Orgun, J. Al-Muhtadi, J. J. P. C. Rodrigues *et al.,* "Design and deployment challenges in immersive and wearable technologies," *Behaviour & Information Technology*, vol. 36, no. 7, pp. 687–698, 2017.

[6] Y. Khmelevsky, X. Li and S. Madnick, "Software development using agile and scrum in distributed teams," in *Proc. of Int. Systems Conf.*, NewYork, NY, USA, pp. 309–317, 2017.

[7] P. Walimbe, "To overcome communication challenges in distributed/virtual scrum teams," *Harrisburg University of Science and Technology*, vol. 12, no. 1, pp. 1–49, 2016.

[8] S. Esquivel, "Communication issues in agile software development," *Journal of Software Evolution and Process*, vol. 1, no. 3, pp. 475–484, 2016.

[9] S. S. Hossain, "Challenges and mitigation strategies in reusing requirements in large-scale distributed agile software development: A survey result," *Journal of Software Engineering Research and Development*, vol. 998, no. 11, pp. 920–935, 2019.

[10] M. Paasivaara, S. Durasiewicz and C. Lassenius, "Using scrum in distributed agile development: A multiple case study," in *Proc. of Int. Conf. on Global Software Engineering*, Otaniemi, Finland, pp. 195–204, 2009.

[11] B. Shahzad and J. Crowcroft, "Trustworthy electronic voting using adjusted blockchain technology," *IEEE Access*, vol. 7, no. 1, pp. 24477–24488, 2019.

[12] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," in *Proc. of Int. Conf. on Software Engineering*, Keele, England, pp. 1–44, 2007.

[13] M. Kasunic, "Designing an effective survey," *Software Engineering Institute*, vol. 9, no. 2, pp. 1–140, 2005.

[14] R. K. Gupta and P. Manikreddy, "Challenges in adapting scrum in legacy global configurator project," in *Proc. of Int. Conf. on Global Software Engineering*, Bangalore, India, pp. 46–50, 2015.

[15] M. Hummel, C. Rosenkranz and R. Holten, "The role of shared understanding in distributed scrum development: An empirical analysis," in *Proc. of European Conf. on Information Systems*, Istanbul, Turkey, pp. 1–13, 2016.

[16] R. Vallon, B. J. da Silva Estácio, R. Prikladnicki and T. Grechenig, "Systematic literature review on agile practices in global software development," *Information and Software Technology*, vol. 96, no. 12, pp. 161–180, 2018.

[17] N. Sekitoleko, F. Evbota, E. Knauss, A. Sandberg, M. Chaudron *et al.,* "Technical dependency challenges in large-scale agile software development," *Journal of Software Engineering Research and Development*, vol. 179, no. 2, pp. 46–61, 2014.

[18] F. Anwer, S. Aftab, S. Shah Muhammad Shah, U. Waheed, S. M. Shah *et al.,* "Comparative analysis of two popular agile process models: Extreme programming and scrum," *International Journal of Computer Science and Telecommunications*, vol. 8, no. 2, pp. 1–7, 2017.

[19] N. Ramadan and S. Megahed, "Requirements engineering in scrum framework," *International Journal of Computer Applications*, vol. 149, no. 8, pp. 24–29, 2016.

[20] M. A. Akbar, J. Sang, Nasrullah, A. A. Khan, M. Shafiq *et al.,* "Towards the guidelines for requirements change management in global software development: Client-vendor perspective," *IEEE Access*, vol. 7, no. 5, pp. 76985–77007, 2019.

[21] J. Al-Muhtadi, B. Shahzad, K. Saleem, W. Jameel and M. A. Orgun, "Cybersecurity and privacy issues for socially integrated mobile healthcare applications operating in a multi-cloud environment," *Health Informatics Journal*, vol. 25, no. 2, pp. 315–329, 2019.

[22] S. Dorairaj, J. Noble and P. Malik, "Effective communication in distributed agile software development teams," *Journal of Software Engineering Research and Development*, vol. 77, no. 2, pp. 102–116, 2011.

[23] D. M. Szabo and J. P. Steghofer, "Coping strategies for temporal, geographical and sociocultural distances in agile global software development: A case study," in *Proc. of Int. Conf. on Software Engineering: Software Engineering in Practice*, Gothenburg, Sweden, pp. 161–170, 2019.

[24] M. A. Razzak and R. Ahmed, "Knowledge sharing in distributed agile projects: Techniques, strategies and challenges," in *Proc. of Federated Conf. on Computer Science and Information Systems*, Stockholm, Sweden, pp. 1431–1440, 2014.

[25] M. K. Mattsson, G. Azizyan and M. K. Magarian, "Classes of distributed Agile development problems," in *Proc. of Agile Conf.*, New York, NY, USA, pp. 51–58, 2010.

[26] S. Shafiq, Y. Hafeez, S. Ali, N. Iqbal and M. Jamal, "Towards scrum based agile framework for global software development teams," *Mehran University Research Journal of Engineering and Technology*, vol. 38, no. 4, pp. 979–998, 2019.

[27] A. Altaf, U. Fatima, W. H. Butt, M. W. Anwar and M. Hamdani, "A systematic literature review on factors impacting agile adaptation in global software development," in *Proc. of Int. Conf. on Computer and Communications Management*, Bangkok, Thailand, pp. 158–163, 2019.

[28] M. Cristal, D. Wildt and R. Prikladnicki, "Usage of scrum practices within a global company," in *Proc. of Int. Conf. on Global Software Engineering*, Porto Alegre, Brazil, pp. 222–226, 2008.

[29] F. Q. B. D. Silva, C. Costa, A. C. C. França and R. Prikladinicki, "Challenges and solutions in distributed software development project management: A systematic literature review," in *Proc. of Int. Conf. on Global Software Engineering*, Porto Alegre, Brazil, pp. 87–96, 2010.

[30] V. Gomes and S. Marczak, "Problems: We all know we have them, do we have solutions too? a literature review on problems and their solutions in global software development," in *Proc. of Int. Conf. on Global Software Engineering*, Porto Alegre, Brazil, pp. 154–158, 2012.

[31] A. Khalid, S. A. Butt, T. Jamal and S. Gochhait, "Agile scrum issues at large-scale distributed projects: Scrum project development at large," *International Journal of Software Innovation*, vol. 8, no. 2, pp. 85–94, 2020.

[32] W. Alsaqaf, M. Daneva and R. Wieringa, "Quality requirements challenges in the context of large-scale distributed agile: An empirical study," *Information and Software Technology*, vol. 110, no. 7, pp. 39–55, 2019.

[33] H. Holmström, B. Fitzgerald, P. J. Ågerfalk and E. Ó. Conchúir, "Usage of scrum practices within a global company," *Information Systems Management*, vol. 23, no. 3, pp. 7–18, 2006.

[34] V. Lalsing, "People factors in agile software development and project management," *International Journal of Software Engineering & Applications*, vol. 3, no. 1, pp. 117–137, 2012.

[35] M. Hossain, "Scrum practice mitigation of coordination challenges in global software development projects: An empirical study," *University of New South Wales in Sydney*, vol. 1, no. 3, pp. 101–179, 2011.

[36] S. Kazi, M. S. Bashir, M. M. Iqbal, Y. Saleem, M. R. J. Qureshi *et al.,* "Requirement change management in agile offshore development (rcmaod)," *Science International*, vol. 26, no. 1, pp. 131–138, 2014.

[37] T. Dingsøyr, N. B. Moe, T. E. Fægri and E. A. Seim, "Exploring software development at the very large-scale: A revelatory case study and research agenda for agile method adaptation," *Empirical Software Engineering*, vol. 23, no. 1, pp. 490–520, 2018.

[38] M. Paasivaara, C. Lassenius, D. Damian, P. Raty and A. Schroter, "Teaching students global software engineering skills using distributed Scrum," in *Proc. of Int. Conf. on Software Engineering*, San Francisco, CA, USA, pp. 1128–1137, 2013.

[39] Z. Wang, "Understanding and managing the challenges of distributed scrum teams," *Blekinge Institute of Technology*, vol. 9, no. 5, pp. 10–59, 2020.

[40] H. Amar, P. M. Rafi-ul-Shan and A. Adegbile, "Towards a 5c theory of communication for scrum-based distributed projects," *British Academy of Management*, vol. 1, no. 3, pp. 25, 2019.

[41] E. Hossain, M. A. Babar and J. Verner, "Towards a framework for using agile approaches in global software development," *Journal of Software Engineering Research and Development*, vol. 32, no. 5, pp. 126–140, 2009.

[42] R. Giuffrida and Y. Dittrich, "A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams," *Information and Software Technology*, vol. 63, no. 7, pp. 11–30, 2015.

[43] M. A. Akbar, J. Sang, Nasrullah, A. A. Khan, S. Mehmoods *et al.,* "Success factors influencing requirements change management process in global software development," *Journal of Computer Languages*, vol. 51, no. 8, pp. 112–130, 2019.

[44] M. Kalenda, P. Hyna and B. Rossi, "Scaling agile in large organizations: Practices, challenges, and success factors," *Journal of Software Evolution and Process*, vol. 30, no. 10, pp. 1–24, 2018.

[45] O. Uludag, M. Kleehaus, C. Caprano and F. Matthes, "Identifying and structuring challenges in large-scale agile development based on a structured literature review," in *Proc. of Enterprise Distributed Object Computing*, Garching, Germany, pp. 191–197, 2018.

[46] R. Amin, I. Inayat, B. Shahzad, K. Saleem and L. Aijun, "An empirical study on acceptance of secure healthcare service in Malaysia, Pakistan, and Saudi Arabia: A mobile cloud computing perspective," *Annales des Telecommunications/Annals of Telecommunications*, vol. 72, no. 5–6, pp. 253–264, 2017.

[47] J. Sutherland, G. Schoonheim, E. Rustenburg and M. Rijk, "Fully distributed scrum: The secret sauce for hyperproductive offshored development teams," in *Proc. of Agile Conf.*, New York, NY, USA, pp. 339–344, 2008.

[48] S. Rahman and D. Arijit, "Mitigation approaches for common issues and challenges when using scrum in global software development," *Blekinge Institute of Technology*, vol. 668, no. 5, pp. 57–117, 2015.

[49] R. Phalnikar, V. S. Deshpande and S. D. Joshi, "Applying agile principles for distributed software development," in *Proc. of Int. Conf. on Advanced Computer Control*, Pune, India, pp. 535–539, 2009.

[50] T. Kamal, Q. Zhang and M. A. Akbar, "Toward successful agile requirements change management process in global software development: A client-vendor analysis," *Institute of Engineering and Technology*, vol. 14, no. 3, pp. 265–274, 2020.

[51] R. K. Gupta, P. Manikreddy and K. C. Arya, "Pragmatic scrum transformation: Challenges, practices & impacts during the journey a case study in a multi-location legacy software product development team," in *Proc. of India Software Engineering Conf.*, Jaipur, India, pp. 147–156, 2017.

[52] K. B. Awar, M. S. I. Sameem and Y. Hafeez, "A model for applying agile practices in distributed environment: A case of local software industry," in *Proc. of Int. Conf. on Communication, Computing and Digital Systems*, Islamabad, Pakistan, pp. 228–232, 2017.

[53] F. Zieris and S. Salinger, "Doing scrum rather than being Agile: A case study on actual nearshoring practices," in *Proc. of Int. Conf. on Global Software Engineering*, Berlin, Germany, pp. 144–153, 2013.

[54] T. Kamal, Q. Zhang, M. A. Akbar, M. Shafiq, A. Gumaei *et al.,* "Identification and prioritization of agile requirements change management success factors in the domain of global software development," *IEEE Access*, vol. 8, no. 3, pp. 44714–44726, 2020.

[55] P. L. Bannerman, E. Hossain and R. Jeffery, "Scrum practice mitigation of global software development coordination challenges: A distinctive advantage," in *Proc. of Hawaii Int. Conf. on System Sciences*, Sydney, Australia, pp. 5309–5318, 2012.

[56] E. Hossain, M. Ali Babar and H. Y. Paik, "Using scrum in global software development: A systematic literature review," in *Proc. of Int. Conf. on Global Software Engineering*, Sydney, Australia, pp. 175–184, 2009.

[57] S. K. Bhatti, M. I. U. Lali, B. Shahzad, F. Javid, F. U. Mangla *et al.,* "Leveraging the big data produced by the network to take intelligent decisions on flow management," *IEEE Access*, vol. 6, no. c, pp. 12197–12205, 2018.

[58] P. Lous, M. Kuhrmann and P. Tell, "Is scrum fit for global software engineering?," in *Proc. of Int. Conf. on Global Software Engineering*, Copenhagen, Denmark, pp. 1–10, 2017.

[59] I. Zada, S. Shahzad and S. Nazir, "Issues and implications of scrum on global software development," *Bahria University Journal of Information & Communication Technology*, vol. 8, no. 1, pp. 81–87, 2017.

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

[61] J. Cho, "Distributed scrum for large-scale and mission-critical projects," in *Proc. of Americas Conf. on Information Systems*, Logan, USA, pp. 399–406, 2007.

[62] F. Almeida, E. Miranda and J. Falcão, "Challenges and facilitators practices for knowledge management in large-scale scrum teams," *Journal of Information Technology Case and Application Research*, vol. 21, no. 2, pp. 90–102, 2019.

[63] E. Therrien, "Overcoming the challenges of building a distributed agile organization," in *Proc. of Agile Conf.*, New York, NY, USA, pp. 368–372, 2008.

[64] M. D. Kahya and C. Seneler, "Geographical distance challenges in distributed agile software development: Case study of a global company," in *Proc. of Int. Sym. on Applied Science*, Samsun, Turkey, pp. 78–83, 2018.

[65] M. Shameem, B. Chandra, R. R. Kumar and C. Kumar, "A systematic literature review to identify human related challenges in globally distributed agile software development: Towards a hypothetical model for scaling agile methodologies," in *Proc. of Int. Conf. on Computing Communication and Automation*, Dhanbad, India, pp. 1–7, 2018.

[66] C. E.Mayer, J. H.Davis and H. D.Foltz, "Texas 5-m antenna aperture efficiency doubled from 230–300 GHz with error compensating secondary," *IEEE Transactions on Antennas and Propagation*, vol. 39, no. 3, pp. 309–317, 2007.

[67] S. Dorairaj, J. Noble and P. Malik, "Understanding lack of trust in distributed agile teams: A grounded theory study," in *Proc. of Conf. on Evaluation and Assessment in Software Engineering*, Wellington, New Zealand, pp. 81–90, 2012.

[68] M.Paasivaara and C. Lassenius, "Scaling scrum in a large globally distributed organization: A case study," in *Proc. of Int. Conf. on Global Software Engineering*, Wellington, New Zealand, pp. 74–83, 2016.

[69] R. Qureshi, M. Basheri and A. A. Alzahrani, "Novel framework to improve communication and coordination among distributed agile teams," *International Journal of Information Engineering and Electronic Business*, vol. 10, no. 4, pp. 16–24, 2018.

[70] K. Dikert, M. Paasivaara and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, no. 7, pp. 87–108, 2016.

[71] M. Paasivaara, B. Behm, C. Lassenius and M. Hallikainen, "Large-scale agile transformation at Ericsson: A case study," *Empirical Software Engineering*, vol. 23, no. 5, pp. 2550–2596, 2018.

[72] A. Välimäki and J. Kääriäinen, "Patterns for distributed scrum: A case study," *Enterprise Interoperability*, vol. 3, no. 1, pp. 85–97, 2008.

[73] B. Shehzad, K. Mehr Awan, M. Ikram-ullah Lali and W. Aslam, "Identification of patterns in failure of software projects," *Journal of Information Science and Engineering*, vol. 33, no. 11, pp. 1465–1479, 2017.

[74] E. Hossain, M. A. Babar, H. Y. Paik and J. Verner, "Risk identification and mitigation processes for using scrum in global software development: A conceptual framework," in *Proc. of Asia-Pacific Software Engineering Conf.*, Sydney, Australia, pp. 457–464, 2009.

[75] M. Shameem, R. R. Kumar, C. Kumar, B. Chandra and A. A. Khan, "Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process," *Journal of Software Evolution and Process*, vol. 30, no. 11, pp. 1–19, 2018.

[76] A. M. AlMutairi and M. R. J. Qureshi, "The proposal of scaling the roles in scrum of scrums for distributed large projects," *International Journal of Information Technology and Computer Science*, vol. 7, no. 8, pp. 68–74, 2015.

[77] A. M. Razavi and R. Ahmad, "Agile development in large and distributed environments: A systematic literature review on organizational, managerial and cultural aspects," in *Proc. of Malaysian Software Engineering Conf.*, Malaysia, pp. 216–221, 2014.

[78] B. Shahzad, K. M. Awan, A. M. Abdullatif, M. I. Lali, M. S. Nawaz *et al.,* "Quantification of productivity of the brands on social media with respect to their responsiveness," *IEEE Access*, vol. 7, no. 1, pp. 9531–9539, 2019.

[79] S. Lee and H. S. Yong, "Distributed agile: Project management in a global environment," *Empirical Software Engineering*, vol. 15, no. 2, pp. 204–217, 2010.

[80] M. M. Jha, R. M. F. Vilardell and J. Narayan, "Scaling agile scrum software development: Providing agility and quality to platform development by reducing time to market," in *Proc. of Int. Conf. on Global Software Engineering*, Pune, India, pp. 84–88, 2016.

[81] M. Paasivaara, C. Lassenius and V. T. Heikkilä, "Inter-team coordination in large-scale globally distributed scrum: Do scrum-of-scrums really work?," in *Proc. of Int. Conf. on Empirical Software Engineering and Measurement*, Lund, Sweden, pp. 235–238, 2012.

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

[83] M. Hammad and I. Inayat, "Integrating risk management in scrum framework," in *Proc. of Frontiers of Information Technology Conf.*, Islamabad, Pakistan, pp. 158–163, 2019.

[84] M. Paasivaara and C. Lassenius, "Scaling scrum in a large distributed project," in *Int. Symp. on Empirical Software Engineering and Measurement*, vol. 49, no. 12, pp. 363–367, 2011.

[85] R. Sriram and S. K. Mathew, "Global software development using agile methodologies: A review of literature," in *Proc. of Int. Conf. on Management and Information Technology*, Chennai, India, pp. 389–393, 2012.

[86] K. Schwaber and J. Sutherland, "The scrum papers: Nut, bolts, and origins of an agile framework," *Journal of Software Engineering Research and Development*, vol. 1, no. 1, pp. 179–224, 2011.