FRAMEWORK FOR EFFECTIVE UTILIZATION OF DISTRIBUTED SCRUM

WARDAH NAEEM AWAN

NATIONAL UNIVERSITY OF MODERN LANGUAGES

THESIS AND DEFENSE APPROVAL FORM

The undersigned certify that they have read the following thesis, examined the defense, are satisfied with overall exam performance, and recommend the thesis to the Faculty of Engineering and Computer Sciences.

THESIS TITLE: FRAMEWORK FOR EFFECTIVE UTILIZATION OF DISTRIBUTED SCRUM

Wardah Naeem Awan

Submitted By:

Registration #:

Master in Software Engineering (MSSE)

Title of the Degree

Software Engineering

Name of Discipline

Name of Internal Examiner

Signature:

Name of Internal Examiner

Signature:

Dr. Basit Shahzad

Name of Research Supervisor

Signature:

Name of Co-Supervisor

Signature:

Dr. Muzafar Khan

Name of HoD (SE)

Signature:

Dr. Basit Shahzad

Name of Dean (FE&CS)

Signature:

August 26th, 2021

"I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of the degree of Master of Science in (*Software Engineering*)"

Signature : ____________________

Name : Assoc. Prof. Dr. Basit Shahzad

Date : August 26rd , 2021

Signature : ____________________

Name : ____________________

Date : ____________________

FRAMEWORK FOR EFFECTIVE UTILIZATION OF DISTRIBUTED SCRUM

WARDAH NAEEM AWAN

A thesis submitted in fulfillment of the
requirements for the award of the degree of
Master of (Software Engineering)

Department of Software Engineering
National University of Modern Languages

AUGUST 2021

DECLARATION

I declare that this thesis entitled "*Framework for Effective Utilization of Distributed Scrum*" is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ____________________

Name : Wardah Naeem Awan

Date : August 26rd , 2021

This thesis work is dedicated to my parents and my teachers throughout my education career who have not only loved me unconditionally but whose good examples have taught me to work hard for the things that I aspire to achieve.

ACKNOWLEDGEMENT

First of all, I wish to express my gratitude and deep appreciation to Almighty Allah, who made this study possible and successful. This study would not be accomplished unless the honest espousal that was extended from several sources for which I would like to express my sincere thankfulness and gratitude. Yet, there were significant contributors for my attained success and I cannot forget their input, especially my research supervisors, Assoc. Prof. Dr. Basit Shahzad, who did not leave any stone unturned to guide me during my research journey.

I shall also acknowledge the extended assistance from the administrations of Department of Software Engineering who supported me all through my research experience and simplified the challenges I faced. For all whom I did not mention but I shall not neglect their significant contribution, thanks for everything.

ABSTRACT

Distributive and Global Software Development (GSD) is the latest on-going trend in software development industry. Global Software Development is the development that is distributed among multiple locations that are separated by nationwide or international boundaries. The COVID-19 Pandemic of 2019 and 2020 has enforced development teams to work in a more distributive manner. 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 software development teams. According to survey the scrum method of agile software development is used over by 89% of agile development teams. Scrum is typically considered to be productive for small-scale projects with co-located teams because Scrum teams are self-organized and enabled on great team collaboration and communication. While, the project stakeholders in GSD projects are usually distributed by time-based, geographic and social and cultural distances that results the generation a numerous challenge or risks that might effect on teams communication and collaboration process. Therefore, we were aimed to identify the challenges confronted by distributed scrum teams and the mitigation strategies adopted by distributed scrum teams to overcome the confronted challenges. 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 distributed scrum experts. 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.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Software Engineering is a method of software development that provide step by step procedure to produce a high-quality software as a result of development process. It ensures the quality of software by ensuring that under development components of the system are planned by the architectural engineers and allow them to enhance the software quality while developing the software. Software Engineering provide various methods of software development with the help of procedures, processes and policies named as software development methodology (SDM). There are numerous software development methodologies having their own life cycle known as Software development life cycle (SDLC). While adopting any SDLC for development, team is actually adapting the policies, procedures and processes of that specific methodology to produce a high-quality software as a result. In actual SDLC alone do not assure the success of any project rather, it supports the team to develop a successful project. There are numerous SDLC models i.e. Spiral, Waterfall and Agile but our research is dedicated for the Scrum model which is further classification of Agile Development [1].

The Agile model is modern and widely used model in the software development. In 2001 it was presented as a whole agile manifesto. The purpose behind presenting this model was to resolve the lacking of effective software development models and to make the development more efficient and easier. In agile, software is developed in various iterations, and it encourage request of changes at any iteration and stage of project development. Customer satisfaction is the main concern of this model, so it involves customer in software development process. The customers are openly engaged in requirement elicitation and development process. Agile model is incomplete without its components, such as scrum. Scrum is the most popular and known component of agile model [2].

Scrum is lightweight framework of agile that provide the direction and steps to manage and control the software development process. Scrum is basically a hybrid model which is the combination of iterative and incremental model [3]. In it software is deliver in different increments called "Sprints" (typically 2-4 weeks iterations). Every sprint starts with planning and ends with a review. Sprint planning is basically a time-boxed meeting from scrum team and could be last for up-to 4 hours. The meeting is dedicated for the development of detailed plans for the sprint. After that, Sprint review meeting is conducted that are 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. 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 team addresses the questions what I have done yesterday, what is my today's task and lastly what obstacles are there to perform my task? Scrum creates 3 artifacts, name as: product backlogs, sprint backlogs and burn-down charts. Backlogs contain requirements form customers and daily burn down indicates the total remaining work [4].

Agile software development methods were initially designed for small scale projects with co-located team members. Co-location allows direct association among team members which results the speedy releases of functioning software [5]. But, the success of agile methods in small co-located projects encouraged organizations to implement these methods in large-scale development [6]. Scrum which is the main component of agile methods has achieved acceptance in recent years and proved that it is quite effective and useful approach for managing projects in many small co-located teams. According to survey the scrum method of agile software development is used over by 89% of agile development teams. It is also stated that scrum could also be used for large scale and distributed projects. Scrum is typically considered to be efficient for small scale projects with co-located teams as scrum teams are self-organized, enabled on great team collaboration and communication, so apparently it is hard to implement scrum practices in GSD because in GSD development team members are separated physically[4] .

GSD is a latest ongoing trend in the software development industry [7]. GSD is software development that is distributed among multiple locations that are divided by nationwide or international boundaries. It provides support to the national organization by providing them the approach to the competent and skillful resources at a lesser cost, accessibility to markets, approach to native knowledge, and freedom in answering to various native opportunities. But, GSD also have some challenges, which is not a part of (or not as prominent) conventional co-located software development projects. GSD usually engage stakeholders situated in various geographic locations and time zones, from different national and organizational cultures, using diverse and, at the same time, unpredictable technologies to cooperate with each other. Such time-based, geographic and socio-cultural distances can leads to major coordination, communication, and control challenges that must be controlled to achieve the benefits of GSD [8].

There is a rising interest of implementing agile practices in GSD projects to get the mutual benefits of both methods [7]. Scrum, as an agile common and most important practice, recently admired by many software teams in agile distributed development. But, the project stakeholders in GSD are distributed by temporal, geographical and culture often leads to the challenges or risks that could affect the processes of communication and collaboration [9]. 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].

Scrum is typically considered to be effective for small team size and co-located projects, as scrum teams are self-organized and enabled on great team collaboration and communication. But, the project stakeholders in GSD projects are usually distributed by time-based, geographic and socio-cultural distances that results the generation a numerous challenge or risks that might effect on team collaboration and communication processes [7].

There are numerous difficulties confronted by many organizations who are using scrum practices. These difficulties are initiated mostly because of geographical distance, cultural differences, communication and collaboration problems, team management issues, information sharing and information management problems, process and organizational supervision issues and practical issues [10]. For this purpose we are determine to identify, discover, examine and describe different challenges and issues that limit the use of scrum in distributed and GSD and to identify mitigation strategies adopted by distributed scrum teams to mitigate the confronted challenges, in order to propose a framework to mitigate and handle the identified challenges [4].

1.2 Problem Statement

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 software development teams. Scrum is typically considered to be productive for small-scale projects with co-located teams because scrum teams are self-organized and enabled on great team collaboration and communication. But, the project stakeholders in GSD projects are usually distributed by time-based, geographic and social and cultural distances that results the generation a numerous challenge or risks that might effect on team collaboration and communication processes. To successfully execute the scrum in distributed environment there is a need to identify, discover, examine and describe various challenges and issues that limit or restrict the use of scrum in distributed and GSD and to propose a framework to mitigate the identified challenges and issues [4].

1.3 Research Questions

Our research composed of following research questions:

RQ1: What are the challenges confronted by teams during the execution of scrum in distributed manner?

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

1.4 Goal of the Study

The goal of our study research is to highlight the challenges that are confronted during the execution of distributed scrum and propose a framework to mitigate or handle the identified challenges.

1.5 Research Objectives

Objective 1: To explore the challenges confronted by distributed scrum team while development.

Objective 2: To propose a framework to mitigate the identified challenges.

1.6 Research Methodology

This section discussed the methods that are used for conducting the research. The first adopted method is SLR, which is conducted to review the existing literature related to the challenges involved in the execution on scrum in distributed environment. The reason of choosing this method is that it provides a broad review and critical analysis of all related existing researches on and around the area of distributed scrum. It is a predefined strategy that helps in conducting reasonable literature review in an organized way. The primary objective of selecting SLR was to come up with the challenges that are confronted by distributed scrum practitioners and mitigation strategies adopted by practitioners to mitigate the confronted challenges. Besides SLR, we conducted an industrial survey and selected as a second research methodology in order to validate the challenges identified through SLR. Lastly, we conducted an expert review to evaluate the effectiveness of our proposed framework. Figure 1.1 shows the research methods used in our study.

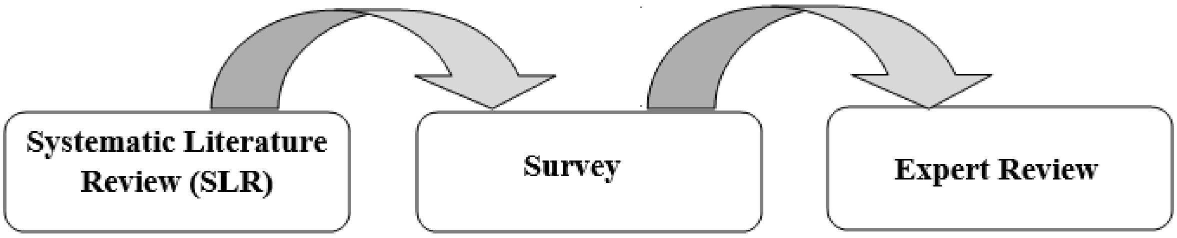

Figure 1.1: Research Methods

1.7 Scope of the Study

The scope of our research restricted to the following:

- This study research is based on distributed scrum rather than traditional scrum.
- The challenges along with their resolution strategies are identified specific to the distributed scrum.
- Conducted SLR of the past 15 year's papers that is from 2005-2020.
- Conducted an industrial survey from distributed scrum practitioners to validate the identified challenges.
- Proposed a conceptual framework to mitigate the identified challenges.
- Conducted expert review to evaluate the effectiveness of proposed framework.

1.8 Thesis Organization

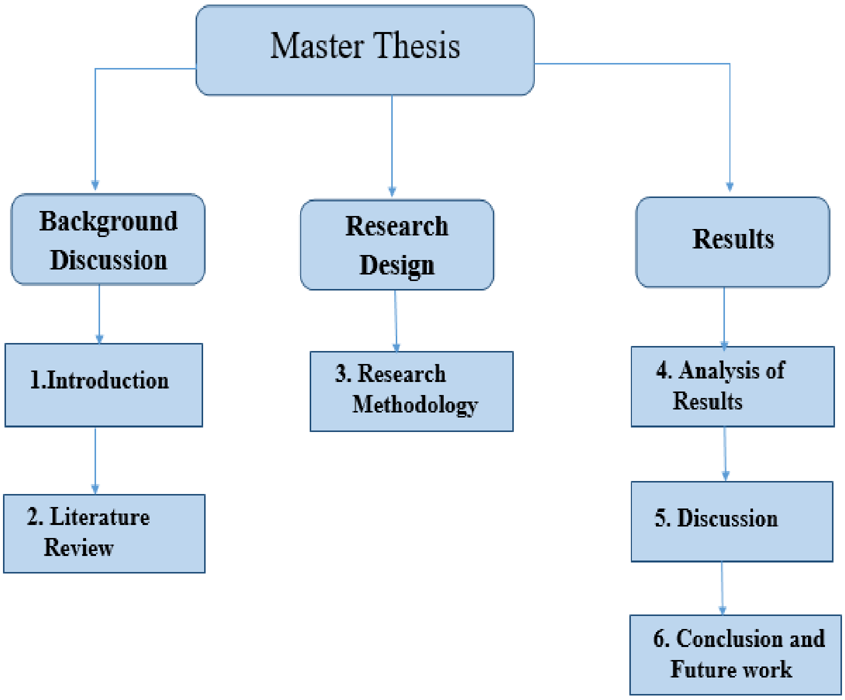

Figure 1.2: Thesis Organization

CHAPTER 2

LITERATURE REVIEW

2.1 Background

Software Engineering provide various methods of software development with the help of procedures, processes and policies named as software development methodology (SDM). There are numerous software development methodologies having their own life cycle known as Software development life cycle (SDLC). While adopting any SDLC for development team is adapting the policies, procedures, and processes of that specific methodology to produce a high-quality software as a result. In actual SDLC alone do not assure the success of any project, rather it supports the team to develop a successful project. There are numerous SDLC models i.e., Spiral, Waterfall and Agile but our research is dedicated for Scrum model which is further classification of Agile development [1].

The Agile model is modern and widely used model in the software development. In 2001 it was presented as a whole agile manifesto. The purpose behind presenting this model was to resolve lack of effective software development models and to make the development more efficient and easier. In agile, software is developed in various iterations, and it encourage request of changes at any iteration and stage of project development. Customer satisfaction is the main concern of this model, so it involves customer in software development process. The customers are openly engaged in requirement elicitation and development process. Agile model is incomplete without its components, such as scrum. Scrum is the most popular and known component of agile model [2].

Scrum is lightweight framework of agile method, has achieved acceptance in recent years and proved that it is quite effective and useful approach for managing projects in many small co-located teams. According to survey the scrum method of agile software development is used over by 89% of agile development teams. It is also stated that scrum could also be used for large scale and distributed projects. Scrum is typically considered to be efficient for small scale projects with co-located teams as scrum teams are self-organized, enabled on great team collaboration and communication, so apparently it is hard to implement Scrum practices in GSD because in GSD development team members are separated physically [4].

2.2 Overview of Scrum Process

Scrum is basically a hybrid model which is the combination of iterative and incremental model [3]. In it software is deliver in different increments called “Sprints” (typically 2-4 weeks iterations). Every sprint starts with planning and ends with a review. Sprint planning is basically a time-boxed meeting from scrum team and could be last for up-to 4 hours. The meeting is dedicated for the development of detailed plans for the sprint. After that, sprint review meeting is conducted that are 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. 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 team addresses the questions what I have done yesterday, what is my today’s task and lastly what obstacles are there to perform my task? Scrum creates 3 artefacts, name as: product backlogs, sprint backlogs and burn-down charts. Backlogs contain requirements form customers and daily burn down indicates the total remaining work [4].

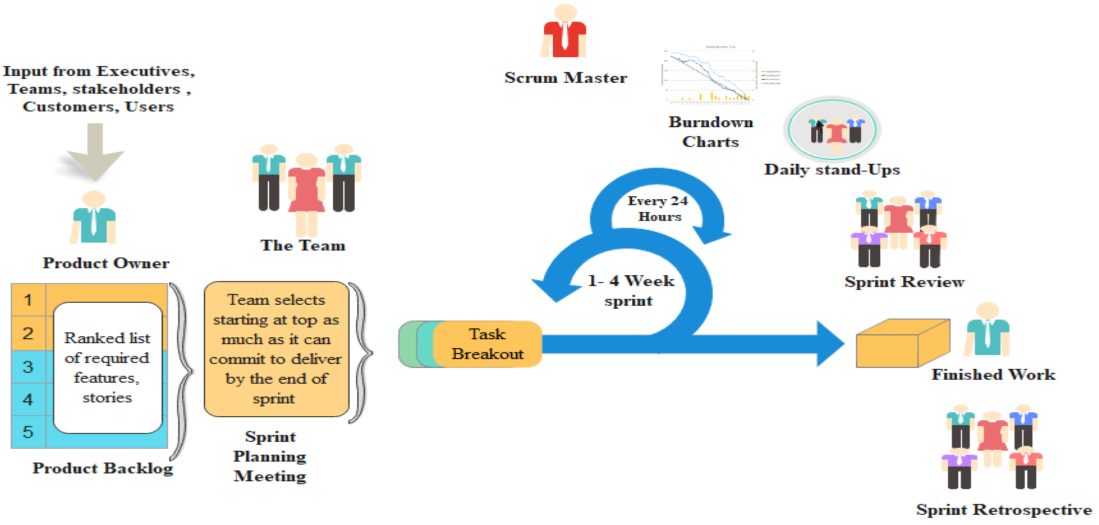

Figure 2.1: Overview of Scrum Process

2.3 Scrum Methodology

The structure of scrum framework is as follows:

2.3.1 Roles in Scrum

In Scrum methodology scrum team is consisted of following three roles: the Product owner, the development team and the scrum master.

i. **The product Owner:** Product Owner is a person who is responsible to maximize the return of Investment (ROI) of product, by finding the features of product, translating those found features into prioritized list, deciding the features for the next sprint and continuously refining and re-prioritizing the list of features. Product owner is responsible for developing, maintaining, explaining and presenting the product backlog to the development team. Product owner is allowed to take decision about the team and the users and is also empowered to accept or reject the final outcomes or results of the work [11].

ii. **The Development Team:** The development team of scrum is self-organized, self-managing and cross-functional. Usually the team size is small in number from seven plus or minus two people, includes people from different professions to develop the potentially shippable product at the end of the sprint [11].

iii. **The Scrum Master:** Scrum Master perform as a mentor for the team to make team follow Scrum practices, processes, theory and rules. Scrum Master is allowed to do whatever is required to make team successful. Scrum master facilitate most of the meetings with team and if team stuck anywhere between the process scrum master is the person who jumps into the situation to resolve the problem [11].

2.3.2 Artifacts in Scrum

Scrum creates 3 artifacts, name as: product backlogs, sprint backlogs and burn-down charts.

i. **Product Backlog:** Product backlog is a stack of prioritized user stories or requirements for the product. Initially it is at very high level and evolve over time. It contains the detail of all the requirements, features, and functionalities a product must have. It is developed and maintained by product owner to maximize the ROI of product [11].

ii. **Sprint Backlog:** In sprint backlog development team selects the list of work by picking up the top priority task from the product backlog and break down the tasks into smaller tasks to develop or complete the task into sprint. The list of tasks that are broken into smaller tasks is known as sprint backlog [11].

iii. **Burndown Chart:** It is the graphical representation of the work progress regarding remaining work versus time. It tell either team will be able to achieve whatever is expecting to achieve or not [3] [11] .

2.3.3 Meetings in Scrum

Sprint is a time-bound typically 2-4 week's long iteration. Every sprint starts with sprint planning and ends with a review of completed sprint [12]. In order to enable the transparency about the progress of project among the team or individuals who are involved in project, different meeting events are conducted in scrum [12].

i. **Sprint Planning**

Sprint planning is basically a time-boxed meeting from scrum team and could be last for up-to 4 hours. The meeting is dedicated for the development of detailed plans for the sprint [4]. Meeting is conducted before starting the sprint and in this meeting, Product Owner and Development team review the product backlog and select the prioritized features form the product backlog to complete them in sprint [11].

ii. **Daily scrum Meeting:** Daily scrum meeting also known as daily standup meeting is conducted by the scrum team and is 15-minute long in which every participant of team address the questions what I have done yesterday, what is my today's task and lastly what obstacles are there to perform my task? [4][11].

iii. **Sprint Review and Retrospective:** Sprint review meeting is conducted after completion of sprint that is attended by the whole scrum team including Product Owner, Scrum Master and development team and the stakeholders of the projects to assess the business conditions, the market, and the technology. The review meetings possibly will be last for more than 4 hours. Retrospective meeting could also be planned to evaluate the teamwork in finalized sprints [4]. Retrospective meeting is about process, not about product. In it scrum team discuss about the process like what went well in process and where the improvement is needed [11].

iv. **Scrum of Scrum Meeting:** When multiple teams are working on same project Scrum of Scrum (SoS) meeting is conducted for effective communication and flow of information sharing among teams [3]. This technique is used to expand large group with more than 12 people into smaller groups or teams and one member from each team will act as ambassador to participate in SoS meeting [11].

2.3.4 Distributed Scrum Model

There are three commonly observed and used model of distributed scrum such as Isolated Scrum, Distributed Scrum of Scrum and Totally Integrated Scrum [13].

i. **Isolated Scrum:** In this model scrum teams are geographically isolated. Mostly offshore teams are not cross-functional or might not be using the process of scrum [13].

ii. **Distributed Scrum of Scrum:** In this model scrum teams are distributed geographically but are connected through regular SoS meetings [13].

iii. **Totally Integrated Scrum:** In this model, teams are geographically distributed and are cross-functional with the members that are distributed among geographies [13].

2.4 Existing Studies related to Challenges in Distributed Scrum

The most recent study was conducted in 2020 but it was particular about communication and co-ordination issues in scrum for large scale teams in distributed and GSD [2]. Distributed Agile Scrum domain is basically chosen because agile approaches have been originally designed for co-located software development, and it is hard to directly apply these methods to distributed development [14], 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. Other studies related to our research is shown in table 1 below:

Table 2.1: Existing Studies on Challenges in Distributed Scrum

Paper No	Author/ Year	Domain/ Paper Type	Limitations	Contribution
1	Ayesha Khalid/2020	Distributed Scrum Issues/ Journal	No proper mechanism is proposed to mitigate the highlighted issues.	Highlighted the communication and co-ordination issues in large scale distributed scrum in GSD.
2	Mohammad Esteki/ 2020	Distributed Scrum Issues/ Journal	Further evaluation of the proposed framework is needed.	Proposed Risk management framework by using PRINCE2 methodology.
3	Imran Ghani/2019	Distributed Scrum Issues/ Journal	Proposed framework is not evaluated.	Proposed a framework of the “Distance Factors and 5-Cs of Challenges” by discussing 4 types of distances that causes 5-Cs challenges.
4	Islam Zada/2017	Distributed Scrum Issues/ Journal	No solution is proposed to resolve the identified challenges.	Highlights the challenges and issues that limit the execution of Scrum practices in globally distributed development.

5	Dr. Hassan Amar/2019	Distributed Scrum Issues/ Conference Proceedings	Detailed understanding of challenges related to communication in distributed scrum is needed.	Discussed the communication related issues in distributed scrum and explore different aspects to improve communication for distributed scrum projects.
6	Fernando Almeida/2019	Distributed Scrum Issues/ Journal	Evaluation of the identified knowledge management practices is required.	key challenges and problems of scaling Scrum practices for large-scale development along with Numerous facilitators' knowledge management practices are identified and discussed
7	Youry Khmelevsky/2017	Distributed Scrum Issues/ Conference Proceedings	Recommended solutions are not evaluated.	Based on a case study several challenges of distributed scrum are discussed and give recommendation to overcome identified challenges.
8	Omer¨ Uluda˘g/2018	Distributed Scrum Issues/ Conference paper	No strategy is proposed to overcome the challenges.	Different challenges are identified and categories them into different groups
9	Markus Hummel/2016	Distributed Scrum Issues/ Conference Proceedings	Collected data is on the individual level	Shared understanding among all the scrum teams is very important to make project successful in distributed scrum
10	Priyamvada Walimbe/2016	Distributed Scrum Issues/ Master's thesis	Lacking in the implementation of specified tools in diverse scrum teams	Effective tools are specified to overcome the communication challenges
11	Maria Paasivaara/2016	Distributed Scrum Issues/ Conference Proceedings	Most of the interviews are conducted on same web site for data collection	Highlights the scaling issues of scrum practices in globally distributed large-scale organization based on case study.
12	Rizwan Qureshi/2018	Distributed Scrum Issues/ Journal	Framework is not validated through any case study.	Communication and Co-ordination issues of DS are discussed, and novel framework is proposed solve the identified issues
13	Paul L. Bannerman/2012	Distributed Scrum Issues/ Conference Proceedings	Only few of the identified challenges are inspected	Co-ordination challenges of DS is discussed, and case study is performed to inspect the challenges.
14	Pernille Lous/2017	Distributed Scrum Issues/ Conference Proceedings	Comprehensive research is missing.	Challenges of scaling scrum in global software engineering is discussed.
15	Emam Hossain/2009	Distributed Scrum Issues/ Conference Proceedings	Comprehensive survey and case study is needed to support the studies.	Identified numerous challenges that limit the execution of Scrum practices in GSD projects.

16	A. Välimäki1/2008	Distributed Scrum Issues/ Book Chapter	Analysis of discussed pattern is limited.	This paper discusses about the distributed scrum patterns and issues related to it and perform a case study
17	Mauricio Cristal/2008	Distributed Scrum Issues/ Conference Proceedings	Scrum needs to be used more systematically.	Performed an industrial study to discuss about the challenges of DS and recommend the solution to overcome the challenges
18	Emam Hossain/2011	Distributed Scrum Issues/ Conference Proceedings	Framework is not evaluated empirically.	Developed a framework to mitigate the challenges in distributed scrum
19	Emam Hossain/2009	Distributed Scrum Issues/ Conference Proceedings	Limited papers are used to propose a framework, so modification is required by reviewing more papers.	Key challenges that limit the execution of Scrum in GSD are identified and consolidate the observed challenges into a framework.
20	Juyun Cho/2007	Distributed Scrum Issues/Conference Proceedings	Unable able to resolve the issues completely that are confronted during development.	Highlights the issues that could arise if DS is used to develop large-scale and mission critical projects and also suggests the possible solutions to resolve the discussed issues.
21	Areej Al-Zaidi/2017	Distributed Scrum Issues/ Journal Paper	Proposed framework is theoretical contribution that needs to be evaluated to find out its validity	Discussed communication challenges in GSD and proposed a framework to overcome the discussed challenges.
22	Abeer M. AlMutairi /2015	Distributed Scrum Issues/ Journal Paper	Further evaluation is needed.	Discussed scaling issues regarding the role of scrum team members in large scale DS and a novel solution is proposed to resolve the scaling issues.
23	Seiyoung Lee/ 2010	Distributed Scrum Issues/ Journal Paper	No specific limitation is found.	Discussed most significant challenges confronted by Yahoo browser while developing latest versions of Yahoo using DS and suggest the practices to overcome the challenges
24	Maria Paasivaara/2009	Distributed Scrum Issues/Conference Proceedings	Results are difficult to generalize, and results are validated by only two organizations.	A multiple case study is performed to find out the successful adoption of scrum practices in distributed manner along with the confronted challenges.
25	Maria Paasivaara/ 2011	Distributed Scrum Issues/Conference Proceedings	Interviews are conducted using only single site.	A case study is performed to discuss scaling issues of DS. Results shows that it is quite tough to practically scale and implement scrum in distributive manner.
26	Nelson Sekitoleko/ 2014	Distributed Scrum Issues/Conference Proceedings	Only single case study is performed.	Different technical dependency challenges in DAD are discussed and found that real benefits of DAD can be

				obtained by mitigating planning and knowledge sharing challenges
27	R. Sriram / 2012	Distributed Scrum Issues/Conference Proceedings	No specific limitation is found.	Literature review is performed and discussed different challenges found in literature and categorize them into 3 broad categories.
28	Jeff Sutherland / 2008	Distributed Scrum Issues/Conference Proceedings	Further evaluation of the model is needed to find out the effectiveness of model.	A case study is performed to discuss different challenges of DS as well as how to achieve the hyper production status in distributive development.
29	Rajeev Kumar Gupta/ 2017	Distributed Scrum Issues/Conference Proceedings	No specific limitation is found.	Different challenges are identified and a case study is performed to discuss how different innovative practices can be adopted to overcome the confronted challenges.
30	Maria Paasivaara / 2012	Distributed Scrum Issues/Conference Proceedings	More empirical studies are required to overcome inter-team coordination issues.	From multiple case study it is found Scrum-of Scrum in not useful to resolve inter-team coordination when number of teams is large.
31	Muhammad Hammad / 2019	Distributed Scrum Issues/Conference Proceedings	More empirical studies are required to perform risk management and to mitigate risks.	Discussed about risk integration management in scrum framework and found that iterative risk management is very important for scrum.
32	Yehia Ibrahim Alzoubi / 2016	Distributed Scrum Issues/ Journal Paper	No specific limitation is found.	Discussed communication challenges and mitigation strategies adopted in an empirical GDAD. Categorize all found challenges into 6 main categories and discuss about the negative impact of all the categories in communication.
33	Martin Kalenda / 2018	Distributed Scrum Issues/ Journal Paper	Further investigation is needed to find out the appropriate solution of the identified challenges	An active research is performed with an organization and discussed about scaling issues of agile methods in large scale organization and the practices that can be used to make company a success.
34	Elaine Therrien/ 2008	Distributed Scrum Issues/Conference Proceedings	No specific limitation is found.	Different challenges and resolution strategies adopted in DS is discussed in this experience report. Form experience it is found that communication is the key for success.
35	Murat Dogus Kahya/ 2018	Distributed Scrum Issues/Conference Proceedings	No strategy is discussed to overcome the discussed issues	A case study is performed to discuss about temporal challenges and found that communication, coordination and

				control are the most challenging factors in temporal development
36	Rajeev Kumar Gupta / 2015	Distributed Scrum Issues/Conference Proceedings	No specific limitation is found.	Performed a case study to discuss about different challenges as well as resolution strategies in DS to build a high performance team for DS.
37	Mohammad Shameem / 2018	Distributed Scrum Issues/Conference Proceedings	Further empirical study of model is needed.	Discuss different critical challenges of DS and develop a hypothetical model to define the relationship among identified challenges and scaling agile methods in GSD
38	Siva Dorairaj School / 2012	Distributed Scrum Issues/Conference Proceedings	Findings of study is limited for specific context.	Grounded study is performed to discuss about the causes of trust issues and its adverse consequence.
39	D´avid Marcell Szab´ /2019	Distributed Scrum Issues/Conference Proceedings	No specific limitation is found.	Discussed a relationship among geographical, temporal and sociocultural distances and agile practices and found that all 3 distances and agile practices are bi-directional and have impact on communication, coordination and control. Different mitigation strategies are also discussed
40	Salvador Esquivel1 / 2016	Distributed Scrum Issues/ Journal	Results of case study is not generalizable	Performed a case study and discussed about different challenges as well as mitigation strategies used by different teams to overcome the confronted issues in DS.
41	Mira Kajko-Mattsson / 2010	Distributed Scrum Issues/Conference Proceedings	Mitigation strategies of all the challenges are not discussed.	Different case studies from literature were analyzed to find out the DAD challenges. Total 13 challenges were found that were further grouped into 6 different classes.
42	Kim Dikert / 2016	Distributed Scrum Issues/ Journal	No resolution strategy is discussed to tackle the confronted challenges, moreover a practical case study is also needed.	Discussed about 35 different challenges found in DS categorized them into 9 different categories. 29 success factors are also discussed and categorized them into 11 different categories.
43	Maria Paasivaara / 2018	Distributed Scrum Issues/ Journal	Scientific evaluation of suggested frameworks by researchers is needed.	Discussed about different challenges and mitigation strategies used by teams to tackle the challenges. Different frameworks suggested by researchers were also discussed.

44	Areebah Altaf / 2019	Distributed Scrum Issues/Conference Proceedings	Data has been collected from only four databases.	Different challenges of DAD are discussed that needs to be addressed to make more organizations enable to adapt agile methodologies in distributed development.
45	Raoul Vallon / 2018	Distributed Scrum Issues/ Journal	Results are not fully evaluated.	Different success and challenging factors of agile methods are discussed and found that more empirical studies are required to make the agile methods more mature and generalizable.
46	Syeda Sumbul Hossain / 2019	Distributed Scrum Issues/Conference Proceedings	Did not found the criteria to categorize the challenges.	Discussed about 14 different challenges in GSD and categorize them into 3 categories. Mitigation strategies related to reusability of requirement in agile methods in GSD is also discussed
47	Wasim Alsaqaf /2019	Distributed Scrum Issues/ Journal	Internal threats to validity for the study is not discussed.	Discussed about different quality requirement challenges, mitigation strategies used by practitioners and the reasons behind the occurrence of challenges in large scale distributed agile development.
48	Mohammad Shameem / 2018	Distributed Scrum Issues/ Journal	Limited sample size were used to validate the challenges.	Different scaling issues of agile practices in DSD are discussed, and found that challenges related to management has the most significant impact on scaling agile practice in DSD.
49	Torgeir Dingsøyr / 2018	Distributed Scrum Issues/ Journal	Longitudinal studies are required.	Discussed about the scaling issues of agile methods in large scale development by conducting exploratory case study for 4 years in Norway.
50	Abbas Moshref Razavi /2014	Distributed Scrum Issues/Conference Proceedings	Limited in scope as it includes the only articles that are published in IEEE	Discussed about different major issues in DS that needs to be resolved such as project planning and management, communication, coordination, collaboration, cultural and quality assurance issues.
51	Madan Mohan Jha / 2016	Distributed Scrum Issues/Conference Proceedings	Suggested strategies to resolve issues are not validated.	Practice study is conducted to find out and discuss about primary challenges of distributed scrum while scaling scrum in 16 globally distributed scrum teams.

52	Khush Bakhat Awar / 2017	Distributed Scrum Issues/Conference Proceedings	Only single case study is conducted to find out the effectiveness of proposed model, so further evaluation is required for effective use of the model.	Discussed different challenges and categorized into 4 major factors. based on identified challenges and already used best practices a theoretical model is proposed to overcome the confronted challenges
53	Emam Hossain / 2011	Distributed Scrum Issues/Conference Proceedings	No evidence is found from paper that the result of conducted case studies are generalizable to other projects.	From empirical multi-case study is conducted and found that collaboration and communication are the main factors that can restrict the use of scrum in distributed development.
54	Helena Holmström / 2006	Distributed Scrum Issues/ Journal	Further evaluation of suggested strategies is required.	From two in-depth case studies in GSD, it is found that geographical, temporal and sociocultural distance are the main factors that can leads to many critical challenges.
55	Shagufta Shafiq / 2019	Distributed Scrum Issues/ Journal	Further evaluation of framework is required.	Different challenges related to communication are discussed and proposed a framework to mitigate the challenges
56	Maria Paasivaara / 2013	Distributed Scrum Issues/Conference Proceedings	Case studies are required to validate the suggested resolution strategies.	Discussed about the challenges faced by the teams while learning GSE practices using distributed scrum

As shown in Table 1 that there is number of research studies conducted in the domain of distributed scrum issues. Like in study[9] by inserting Scrum in PRINCE2 methodology a risk management framework was proposed. Risk factors are found along with five different categories of software development which includes lifecycle, project management, collective awareness, collaboration with external stakeholders and lastly the technology launch. Outcomes of the framework Implementation revealed that the software development life cycle, project management and collective awareness were the riskiest categories of the study. The limitation of this research is that it is evaluated using one case study only so more evaluation is needed to check out the effectiveness of the framework.

This research [5] is SLR that focused on challenges faced by the distributed agile approach at large scale. In it five types of key challenges were observed that affect the SDLC phases within distributed agile development at large scale and named as the "5-Cs of Challenges":

Communication, Coordination, Cooperation, Collaboration and Control. Four types of distances were also identified that contributes to the existence of these 5 types of key challenges. These distances are geographical, time-based, social and cultural. Based on these, “Distance Factors and 5-Cs of Challenges” framework is proposed. It is acknowledged that there are very few newly suggested solutions that could address the challenges in distributed environment.

Two papers are found that discuss about the benefits and challenges of distributed scrum [10] [14]. Paper [10] identified and discussed the benefits of using distributed scrum like: Reduce Ambiguity, Understanding deficiencies, Facilitate coordination, Maximize stability etc. It also discovered the various challenges that restrict the execution of distributed Scrum practices globally like: ineffective use of SCRUM practices, Physical distance, Cultural differences, Team management, Communication and collaboration issues, Risk management, Different working hours, Knowledge management, Inexperience, Technical issues, Time Delays, Technical Challenges, Data Privacy challenges, and Infrastructure challenges. The limitation of this paper is that it does not suggest any solution or strategy to resolve the challenges. Second paper [14] perform case study that discussed the some challenges and benefits related to distributed scrum. This paper contributed by providing practical recommendations, which could be used by other organizations that are considering adopting the Scrum practices in distributed projects.

Three papers are found about communication challenges in distributed scrum[15][16][17]. In one of the paper, the issues and challenges related to communication in distributed scrum are reviewed and identify the factors that can be used to improve communication for distributed scrum projects. The limitation of this paper is that Detailed understanding of challenges related to communication in distributed scrum projects is missing [15]. While in other paper communication issues are discussed and the effective tools are recommended to overcome the communication challenges. But the suggested tools are not evaluated in distributed scrum teams [16]. In paper[17] author discuss about the communication challenges and some other factors that are effecting the communication and coordination that might restrict the use of scrum in distributed manner. A theoretical framework is proposed to mitigate the confronted communication issues, but the evaluation of framework is needed to find out the validity of the framework. This paper also highlights the benefits of using scrum in distributed manners.

Knowledge management and Sharing is a vital source for the success of scrum teams. However, commonly recognized knowledge management practices usually discuss only team level collaboration, which is quite hard to scale to multiple Scrum teams. For this purpose an empirical study is conducted and paper [18] identify and discuss about numerous facilitators' knowledge practices , and also discuss the key challenges and problems of scaling Scrum practices for large scale teams.

This paper [6] identify the distributed scrum challenges from literature. Since scrum approaches are initially designed for small teams, unusual challenges arise while presenting them at larger scale, such as within team communication and coordination, general resistances to changes or dependencies with other organizational units etc. Total 79 challenges are identified that are categorized into eleven groups. The limitation of this paper is that no strategy is proposed to overcome the identified challenges.

Researcher in paper [19]conducted a field study in a software product development company in order to investigate the role of shared understanding in the distributed Scrum development process. From study it is observed that shared understanding is an important success factor of distributed project success when developing according to Scrum. Intuitions of study is based on quantitative and qualitative data obtained in a field study of a software product development company. Result shows that when following certain strategies, team distribution may not certainly be stopping factor for shared understanding and project success. The limitation of the study is that data used in study is collected on individual level.

Two papers were found in which research was about challenges of scaling scrum in distributed environment [20] [21]. In paper [20] researcher performed a case study in an organization and point out the significant challenges that were faced during case study performance despite of using Large-scale Scrum (LeSS) framework. Organization performed experiment using numerous ways of implementing scaling practices like implementing Scrum-of-Scrums meetings, common sprint planning meetings, common retrospectives, and common demos, and also scaling role of Product Owner. Data used for case study was collected through interviews and most of the interviews were conducted on same site which is the limitation of study as interviewing additional people using multiple sites might have given further perceptions. In [21] different solutions are structured and mapped on various identified challenges from literature, and contributed a great number of strategies and practices that helps to scale the Scrum in distributed context. In this limited literature were used so comprehensive study is needed to evaluate the effectiveness of proposed strategies and practices.

Three papers were found related to the communication and co-ordination challenges in distributed scrum [22][8][23] . In paper [22] communication and co-ordination(C&C) issues are discussed in distributed scrum and proposed a novel framework to solve the identified issues. Questionnaire is used for the validation of proposed framework. The result shows that the framework would be helpful to overcome the C&C issues efficiently and effectively, but the further evaluation of framework is required by using case study to find out its effectiveness. While in paper [8] author discussed the co-ordination challenges of distributed scrum and a case study is performed to examine the challenges. It is identified that Scrum offers a particular benefit in justifying physical and socio-cultural distances but not time-based GSD coordination challenges. Total 12 challenges were identified and only four of them inspected, remaining challenges need to be investigated. The main contribution of this research paper was to highlight the unique characteristics of the Scrum method that can provide benefits to the both distributed as well as co-located software development projects by justifying physical and social and cultural distance based coordination challenges in GSD. In paper [23] literature review is conducted and a research framework is proposed. The framework shows recent knowledge and opinions on in what manner Scrum practices could be used to resolve usually identified communication, co-ordination, and control challenges in GSD. Limitations of the paper is that the proposed framework is not evaluated through any empirical study, so it is just a theoretical contribution.

Four papers were found that discuss about various challenges that may limit the execution of scrum in distributed development[4][24][7]. In paper [4] researcher focus on the identification of various challenges that limit the execution of Scrum practices in distributed and GSD. Another focus of the research was to explore the potential strategies to resolve the identified challenges. Findings of the paper is that there is an immediate need to perform more qualitative and quantitative empirical studies to explore, describe, explain and evaluate the usage of scrum in GSD projects. Detailed case studies and comprehensive survey is also needed to support of the usage of scrum practices in GSD projects. In [24] author presented the experience on the usage of scrum practices in two experimental projects within a global company and write up the experience report. Report shows that the usage of scrum in these two experimental projects was very challenging as the company's culture was not agile-oriented, and it was quite hard to change the mind sets of people for using scrum with already in use traditional practices. But the initial results were very encouraging, and there is a need to use scrum in a more systematic way to get better results. Paper [7] identify the key challenges that limit the use of Scrum in GSD. From literature, seven broad classifications of challenges are identified and then categorize the identified challenges into different GSD issues, but the categorization is not complete yet that is the limitation of the paper. After that the findings are combined, and a conceptual framework is proposed and discuss different components of that framework. The framework is developed based on limited papers so in-depth industry-based case study is needed to validate the framework.

This paper [25] discussed the patterns and the challenges related to the development patterns that are used in distributed scrum development. Different challenges were identified, and a case study is performed. The outcomes of the case study shows that distributed scrum has many benefits if it gets implemented successfully, and the members of 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. Limitation of the paper is that the analysis of proposed patterns is not performed.

Paper [26] perform a case study and highlight the key issues that could arise while developing large-scale and mission critical system by using distributed scrum such as information and knowledge sharing issues, communication and coordination issues, trust and confidence issues, control issues and training issues. It also discusses the mitigation strategies that could be used to resolve the discussed issues that are confronted while performing case study. The limitation of this paper is that the company is unable to completely resolve the confronted issues and there are still some issues that company needs to resolve to make distributed scrum sustainable for the large-scale and mission critical projects.

In paper [27] author discuss the scaling issues regarding the role of scrum members in large scale distributed scrum projects. Scrum of Scrum which is the enhancement of simple scrum is used to resolve the scaling issues by dealing with large scale projects having larger teams. A novel solution is proposed by using Scrum of Scrum method and a survey questionnaire is used to evaluate the proposed solution. Result shows the effectiveness of the proposed solution, but further evaluation is required by performing any case study to find out the validity of the proposed solution.

Paper [28] performed a case study on My Yahoo to find out the performance of distributed development by using distributed waterfall and distributed agile methods specifically by using scrum methods. It is found that the satisfaction and overall performance of the latest versions of My Yahoo that are released by using distributed scrum methods is increased by 30%. Author also discussed the Challenges confronted while developing through distributed scrum it is found that communication, control, and trust issues are the most significant challenges. Moreover, the practices are also discussed that could be used to overcome or control the confronted challenges.

In [29] a multiple case study is performed by involving 3 different small and medium scale organizations who are performing development through distributed scrum. A semi-structured interview is performed to find out the how scrum practices are successfully adopted in distributed manner and GSD supporting practices are also discussed. It also discusses the benefits and the challenges reported by case study and it is found that communication is the main challenge in as scrum expects the open communication. This case study also discusses different lessons learned while performing case study. The limitation of this case study is that as it is a multiple case study, so it is difficult to generalize the results of case study. Moreover, among three organization involved in case study the data was validated by only two organizations.

Paper [30] performed a case study on the organization which is using scrum practices and the size of team in increased by 2 collocated teams to the 20 distributed teams within 2.5 years. A semi-structured interview was conducted to find out the challenges of scaling scrum by interviewing project personals including architects, managers, testers and developers. 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 the most of interviews were conduct from main site so there are chances of biasness as results might be different if other sites have been involved.

In paper [31] a case study is performed to find out the technical dependency challenges in distributed agile development. Different challenges are found such as planning, knowledge sharing, task prioritization, code quality and integration. Followed-up questionnaire is performed to find out the relationship among identified challenges. It is found that if one of the identified challenges occur it is most likely that other challenges will be occur and became problematic by making vicious circle as occurrence of one challenge leads to the occurrence of many other challenges. It is also found that mitigation one of the identified challenges have positive effect on other challenges that leads to the break of vicious circle. Results of case study revealed that by paying attention on mitigating the planning and knowledge sharing challenges will enable effective communication across teams, leads to get the actual benefits of distributed agile development.

In this paper [32] a literature review is conducted to find out the challenges of implementation of agile in distributed manner. Different challenges are found form literature and categorize all the challenges into 3 broad categories such as Governance related issues, Performance of Global software development and software engineering process issues. It is found that issues related to governance and process require special attention to get resolved as these issues has direct influence in distributed development.

In [33] author discuss that whether it is possible for development teams to achieve hyper productive state consistently when shifting from collocated to fully distributed environment. For this purpose, Xebia ProRail PUB project case study is performed where teams are doing development by using XP methods in distributed scrum model. Different advantages that were realized during the development is discussed as well as challenges that were confronted during the implementation of fully distributed scrum is also discussed. It is found that the model that is used in the development of case study project is very helpful for increasing the productivity of the developers who are experienced in doing development using agile methods. The validity of the used model is further needed as only single case study is performed using this development model.

In [34] a case study is performed and discuss the challenges that are confronted by the team as team is transforming form traditional development to distributed agile-scrum development. Different challenges are faced by teams such as collaboration, communication, technical debt, testing and scaling business. It also discussed that how already used process can be evolve by using different innovative practices like collaborative events, new taxonomy for Scrum roles and responsibilities and innovative dashboard to overcome or resolve the confronted challenges.

In scrum literature Scrum-of-Scrum method is discussed but how it is applied in large scale globally distributed projects so paper [35] discuss about implementation of Scrum-of Scrum to resolve inter-team coordination issues in globally distributed project with large number of teams. Through multiple case study, it is found that scrum-of scrum is useful only when number of teams are less as it is not possible for the representative of all team members to answer all the question in just 15 minutes. Two cases which is studied in this paper use different models to resolve the issues, but they are unable to overcome the issues even by using those models. By conducting 58 semi-structured interviews from project personnel, product owners, project managers, architectures, developers and testers, it is found that there is a severe need to perform more empirical studies to overcome the inter-team coordination issues in large scale distributed teams.

Paper [36] discuss about risk integration management in scrum framework. Different risk were identified and the model that were used for risk integration is validated by performing controlled experiment on the semester project of under graduation students. Significant number of critical risks were identified in later sprints which shows that an iterative risk management is very important in scrum. It is found that very little amount of work is done in risk management in agile development and more empirical work is needed to mitigate the risks in agile development.

In paper [37] SLR is performed to discuss the communication challenges and the mitigation strategies applied in empirical globally distributed agile development (GDAD). This study identifies a number of challenges that need to be addressed for efficient and effective communication in GDAD. Different challenges were identified and categorized into six categories such as Team Configuration, Distance Differences, Project Characteristics, Customer Communication, Human Factors and Organizational Factors. Each category contains some challenges regarding communication. It is found that distance difference category is the most discussed category in literature including time zone and geographical distance in GDAD among teams. The negative impact of communication challenges in GDAD and the techniques to mitigate those negative impact is also discussed in this study.

In [38] author discuss about the identification of challenges, success factors and practices used in scaling agile in distributed manner. An action research is performed with the large-scale company. Different challenges were identified, and it is found that company is facing the same challenges as identified in literature. The most critical challenges that company faced is resistance to change, quality assurance issues, fast roll-out and integration of previous and non-agile parts of organization. When research is performed with case company, it is found that there are different factors that can helps to make that organization success such as unification of view, already adopted agile culture and previous agile and lean practice experiences and management support. It is argued that there is a need to deeply investigate the identified challenges to find out the appropriate solution of the challenges.

In this [39] experience report author discuss about the challenges faced by the distributed scrum teams working under the First American Core logic (FACL) even though the teams are trained for scrum process. Key challenges that were found are time zones, communication and cultural, trust issues and technical challenges. Different short- and long-term solutions were also discussed to overcome the confronted issues. From experience it is found that communication is the key to make project successful.

This paper[40] discuss about the challenges associated temporal distance in distributed software development that might be risky by considering current software development industry. A case study is performed with three small scale, one medium scale and one large scale organization to find out the temporal challenges in distributive development and detailed interview were conducted from 12 participants working with German-based Company which is operating in more than 20 countries all over the world. It found that communication, coordination, and control are the most challenging factors in temporal distance development.

In [41] a case study is performed with Global Configurator Project (GCP) to find out the challenges that might be confronted while adopting distributed scrum. Three key challenges are discussed while adoption of distributed scrum such as communication and coordination, team collaboration issues and knowledge sharing issues. Different strategies that were adopted to overcome the issues while performing case study is also discussed such as for improving team collaboration Scrum-of- Scrum is conducted daily, or bi-weekly or on weekly basis to ensure the transparency of the process and to build the high performance teams. Similarly different strategies for overcoming and improving communication and knowledge sharing issues are also discussed.

In [42] author conducted SLR to find out the human related factors that can influence negatively while adopting agile in GSD. From SLR 11 different challenges were found and 6 out of 11 challenges found critical challenges having the frequency of ≥ 50%. The frequency is found by using specific critical challenges criteria of scaling agile methods in GSD. The challenges that found critical were lack of communication with highest frequency of (88%) Lack of customer involvement having 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%). Based on identified challenges a hypothetical model was developed which shows the relationship among identified challenges and scaling agile in GSD.

In [43] grounded theory is performed by involving 45 practitioners form 28 different companies to find out the concerns that could affect the distributed agile development. Different issues are found that need to be addressed to get the benefits of distributive agile development. Author reported that trust is an important concern that effect the distributed development and address the causes of lack of trust such as No sense of belonging, Sense of vulnerability, Poor team bonding, Lack of cultural understanding, Missing face-to-face interaction and Ineffective communication and its adverse consequence on development such as Lack of commitment, Ineffective Collaboration, Team conflict and poor team performance. It is stated that for building trust among teams there is a need to understand the causes and consequences of lack of trust among teams that eventually will pave the path to the successful distributed agile development.

In this paper [44] a case study is conducted to explore the challenges associated with temporal, geographical and socio-cultural distances and to find out the relationship among theses 3 distances in GSD. By interviewing six practitioners form both offshore and onshore teams it is found that all these 3 distances and agile practices are bi-directional, and both have effect on each other. All 3 distances lead to coordination, communication and control issues in GSD. Different agile and non-agile strategies were discussed that were used to cope to issues confronted while performing the case study as author stated that only pure agile practices were not enough to resolve the confronted challenges so other strategies are needed.

This paper [45] discuss about the communication challenges while implementing scrum in GSD by studying different case studies which has implemented the scrum in distributed manner highlight different challenge that were confronted by teams as well as the resolution strategies that were adopted to overcome the confronted challenges. It is stated that for better communication among team or to enhance communication there is a need for the availability of technological infrastructure with many other factors.

In this paper [46] author analyzed 12 different case studies from literature to find out the challenges faced by teams while doing distributive development using agile methods. Total 13 challenges were found that were categorized into 6 different classes such as communication, collaboration, trust issues, cultural issues, temporal issues and training and technical issues. Different strategies are also discussed that were adopted by different teams to overcome the issues.

In [47] author conducted an industrial SLR of large scale agile transformation to find out how agile methods and lean practices of software development are adopted at large scale distributive development. From SLR 35 different challenges were found that were face by different teams and then further categorized into 9 different categories such as Resistance to change, Difficulty to implement agile, Lack of investment, Coordination challenges in multi-team environment, Emergence of different approaches in multi-team environment, Organizational boundaries and hierarchical management, Quality assurance issues, Requirement engineering challenges and Integration of non-development function. Moreover 29 different success factors are also discussed that were further categorized into 11 categories. The most significant success factor among all 11 categories were training and coaching, mindset and alignment, management support, and choosing and customizing the agile model. No specific strategy is discussed to tackle the identified challenges and no case study is found which discuss the transformation of agile methods in large scale distributive development.

To fill the gap that were mentioned in paper [47] paper [48] conducted a case study by introducing agile methods in Ericsson company by developing a XaaS platform. Case study is based on 45 semi-structured interviews and 5 observation session on 3 different sites. In this study author discuss all the challenges face by teams while transforming development to agile methods. The challenges that were faced by teams are Resistance to change, Technical debt issues, Lack of agile training, Lack of common agile framework, Lack of coaches and coaching, Cross-site teams, Lack of continuous integration and test automation, Any team cannot implement any feature, Challenges in breaking down the requirements, Challenges in defining in product owner roles, Constant change and Backlog challenges. Mitigation strategies are also discussed that were adopted by teams to tackle the challenges, but there are some challenges whose mitigation strategies are not discussed and still need to be addressed. Moreover, form literature it is found that the scaling frameworks that are suggested by the consultants such as LeSS, SAFe and DAD have almost no scientific study about their usage and suitability regarding different environments, that need to be performed to use these frameworks.

In paper [49] author conducted a SLR to find out the factors that have significant effect on the adaptation of agile methods in globally distributed development. Different success factors and challenges were found and discussed. It is found that agile methodologies are highly adapted by different organizations because of its iterative models and quick delivery, but there are some basic challenges that need to be addressed such as communication and coordination issues, lack of customer involvement, lack of documentation, project complexity, temporal issues, neglecting NFR and inappropriate architecture as these challenges are badly effecting the growth of software development at global level. Further work is required to handle and overcome the confronted challenges so that more organization will be able to adapt agile practices in global software development.

In [50] SLR is conducted to find out the agile practices used in GSD. From literature it is found that form last few years scrum became the most frequently used process for distributive development. Different case studies are also reviewed and the challenges and the success factors that were identified and discussed in case studies and literature is also discussed. Different challenges are found form literature that were further categorized into main categories by authors. The identified challenges are project management challenges, communication, coordination and control challenges, trust issues, temporal issues, training and technical issues and cultural issues. Different strategies are also discussed that were used and discussed by different authors in their work. From research it is found that there is a serious need to perform more fully empirical studies in this field to make agile methodologies more mature and generalizable so more organization would be able to use it more effectively.

In paper [51] author discuss about the challenges associated with the reusability of requirements in globally distributed agile methods and the mitigation strategies are also discussed suggested by the practitioners. A survey is conducted to identify the challenge related to reusability of requirements and total 14 challenges were identified that were further categorized into communication, coordination and control processed in the perspective of GSD. Some common challenges found from literature as well as by interviewing the practitioners while conducting survey. The common challenges were lack of communication and coordination among dispersed teams, same level of skills is required to reuse the code, lack of awareness of the roles of team members, trust issues for sharing and reusing the code, lack of detailed requirements and architectural issues such as inconsistent platforms or architectural designs. By interviewing practitioners, it is found that different challenges still need to be addressed such as team awareness issues, security and trust issues and architectural issues. Limitation of paper is that it stated that 14 challenges and 10 mitigation strategies are discussed but we do not find the reported number of challenges and mitigation strategies. Moreover, it is not clear that which criteria is used to categorize all the challenges into communication, coordination and control challenges.

In [52] a multiple case study is performed and 15 different challenges were identified that were confronted by the practitioners during case study while dealing with quality requirements in large scale distributed agile development. The confronted challenges were categorized into 5 categories such as communication and coordination challenges among teams, quality requirements elicitation challenges, quality assurance challenges, architectural challenges and conceptual challenges of quality requirements. Moreover the 13 mechanisms that were causing the challenges are also identified and discussed and 9 different mitigation practices that were being used by practitioners to lessen the impact of challenges is also discussed. From this study it is also found that while applying any strategy to mitigate the challenges can also introduce some new challenges so practitioners should analyze mitigation practices properly before using it.

In this paper [53] literature review is conducted to explore and prioritize the challenges regarding scaling practices of agile in distributed software development (DSD). The study was conducted into three steps, firstly literature review is conducted and total 22 challenges were identified that are categorized into 4 categories such as process, team, technology and management. Then on 2nd step to validate the identified challenges an online questionnaire is conducted. 58 responses were collected and on the basis of analysis of the responses challenges get validated. On last step an Analytical Hierarchical Process (AHP) is used to prioritize and categorize the identified challenges. It is found that management is the most important category that could affect the scaling of agile methodology in DSD as management category include challenges such as lack of effective communication, lack of management commitment and the challenges due to temporal differences.

This paper [54] discuss about an explanatory case study conducted in Norway for 4 years on one of the largest software development programs to find out how agile methods are evolved over time to use for large scale and distributed development as agile methods are consider to be suitable for small collocated project 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 [55] SLR is conducted to find out 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 needs to be addressed and resolved to effectively execute agile methodology in distributive manner. This paper limits in a way that it includes the only articles that are published in IEEE, so the findings of this paper are limited in scope.

This practice study discusses about the challenges or issues that were encountered while scaling distributed scrum in total 16 teams that were globally distributed. The issues encountered by distributed scrum teams are: temporal issues, communication, coordination and collaboration issues, knowledge management issues and cultural issues[56].

In paper [57] systematic mapping is performed to find and discuss about the critical factors that affect the implementation of agile practices in distributed manners. Different challenges were found that are categorized into four major factors such as organizational, management, process, and human factor. Theoretical model is proposed based on the identified factor and the best practices used to overcome the confronted issues. To find out the effectiveness of proposed model, case study was conducted for 55 days, with the team that were geographically distributed in different countries such as Pakistan, Turkey, and USA. It is found that the proposed model is suitable for teams to get the benefits of agile practices in distributed manners. Limitation of this study is that only single case study is conducted to find out the effectiveness of proposed model and further evaluation is needed for the effective use of proposed model.

In paper[58] an empirical research based on multi-case study is conducted in order to investigate the scrum practices that are being used in GSD and to identify the key factors that can impact and influence the adaption of scrum in distributed development. Case study was based on four projects including PaperInfo, EnergyInfo, CollaborationSoft and TestSoft and found that collaboration and communication challenges that occur due to social, cultural, temporal and geographical distance are the main factors that can limit the use of scrum in distributed manner. From multi-case study results it is found that by focusing and considering the context of projects and by using the appropriate communication and collaboration tools and mechanism, scrum can be adapted successfully in distributed development. Limitation of this papers is that no evidence is found from the paper that the result of conducted case studies can be generalizable to other projects.

In [59] author discuss about the critical challenges faced by teams while working in GSD environment. It is found that geographical, temporal and sociocultural distance enforce several challenges like communication, coordination and collaboration issues among distributed teams, difficulties regarding knowledge management and mutual understating among distributed teams. Two in-depth case studies were performed and revealed that scrum would be an effective approach to reduce the challenges related to GSD.

In paper [60] author discussed that from literature, it is found that agile methodologies can be effectively used to mitigate the problem associated with distributed development. For that there is a need to figure out the techniques and strategies to effectively execute and integrate the scrum practices in distributed environment. From literature, it is found that due to geographical distance and distributive environment, there is a lack of effective communication among stakeholders and team members leads to different challenges such as inappropriate allocation of tasks, lack of management for early detected faults, and exploration of limited resources. So in this papers framework is developed to mitigate the scaling issues of scrum in distributive manner and to optimize the performance of scrum in distributed development. To validate the effectiveness of framework theoretical and empirical evaluation is performed, and the results revealed that proposed framework is effective to minimize or mitigation the issues and challenges associated with geographical distance.

In paper [61] a course design is discussed that has adopted distributed scrum in global software engineering (GSE) to teach the globally distributive software engineering skills to the students. 16 students involved in the course from Canada and Finland. Mixed method research is used to assess the GSE learning of students. It is reported that communication is the key challenge that cause many other challenges such as trust issues, temporal, cultural, and motivational issues. From analysis results, it is found that supporting practices of distributed scrum such as collaboration and tools supports can help to learn the important competencies of GSE and to mitigate the challenges.

2.5 Summary

In this chapter we have discuss the background and related work of our research in detail. Background is discussed to give brief insight into scrum process. We conducted in-depth systematic literature review by following Kitchenham guidelines and reviewed 56 papers from year 2005-2020. The purpose of SLR was, to identify the challenges confronted by distributed scrum teams and mitigation strategies adopted by teams to overcome the confronted challenges.

Findings from each study were extracted and placed into a tabular from for better understanding shown in table 2.1.

CHAPTER 3

RESEARCH METHDOLOGY

3.1 Overview

Different methodologies can be used for conducting research depending upon the type of research which is either exploratory or descriptive or explanatory. Quantitative and qualitative methods are two of them that are being used for conducting research. These two methods are totally different in nature from each other but at the same time tightly coupled with together. Quantitative methods focused on statistical data or the data that is measurable and could be used for larger number of cases, people or groups of people. So generally, quantitative method translate data into the numbers or statistics and adds understanding in information. While qualitative data focused on the detailed description of data and used for small number of cases, people or groups. It is represented in words rather than numbers unlike quantitative method. It is also feasible to combine both quantitative and qualitative methods, like for data collection qualitative methods can be used, and for the analyses of data quantitative method can be used. The method of combining both methods is called mixed methodology[62].

3.2 Quantitative Research

Quantitative research method is the numerical or statistical representation and explanation of the observed data. Normally, quantitative is all about collecting the numerical data for the explanation of particular situation or to answer the particular question, like to find out the percentage of the understanding of students for particular subjects. Quantitative method in general focused on measuring the social reality and establishing the research numerically. It is used when researcher wants to compute the opinions and the behavior of people about some certain situation something. It is also used to find out the ratio of some certain situation by segmenting the audience into different groups, and for that population with similar opinions is placed in one group while population with distinct opinions is placed in another group. This method is most suitable for testing the hypotheses or when some situation or phenomena is needed to explain. Different types of quantitative methods are available such as survey, experiment and action research[63].

3.3 Instruments of Quantitative Research

Following instruments are used to conducted quantitative research:

3.3.1 Survey

Survey is a method of data collection and analysis and is an extensively used approach in software engineering as it could be used to depict the attitudes, behaviors and the information of a large group of people by studying their subset. It helps in decision making by providing the understanding of complex problems, and by suggesting the solution for the problems [64]. In this method data is mostly collected by conducting standardized questionnaire so, it is closely associated with questionnaire. But survey research could also be conducted by conducting interviews, and interviews could be either structured or semi-structured un-structured. Data is collected by getting the answer to already prepared questions from respondents. Questionnaire consist of number of question written in a definite order. Questionnaire is sent to the concerned person either by post or emailed them with the request to fill the questionnaire and send it back. If survey is conducted through interview, then in-person or telephonic interview can be conducted. Preliminary condition for conducting a survey is a clear research question and the population that will be targeted, as it is not possible to collect the required data without any targeted population and moreover it would be either unfeasible or unnecessary to collect the data from whole population. So firstly, it is a needed to identify the targeted population sample or representative subset of the population as a sample and then how to reach that sample for collection of data is needed to determine. Different methods are used for selecting sample depending upon the type of research. Probability and non-probability sampling are two broader types of sampling and different methods of sampling lies under these two types such as random sampling, systematic sampling and quota sampling etc. After collection of data, analysis techniques are used to make data generalizable for the population [65].

3.3.2 Experiment

Experiment method is used when researcher wants the control over variables and the situation. An experiment method is used to find out the causal relationship among one or more dependent and independent variables. Controlled experiments are used to determine whether the causal relationships exist between variables and how they are related to each other. Independent variables are manipulated to find out its effects on dependent variables. Causal relationship is used to identify and explain the reason behind the occurrence of any event, while the determination of causality processes provides an explanation of how a phenomenon occurred. For every independent variable, combination of values is a treatment. Every experiment contains at least one treatment consists of an attribute for treatment, unit of assignment, measure of outcome and the comparison to infer the change.

Preliminary condition for conducting an experiment is a clear hypothesis. Experiments are theory driven as hypothesis is drawn from theory. Hypothesis and the theory from which hypothesis is drawn is a guide to perform all steps of experiment including the decision of variable which will be used and how to measure those variables. Theory helps to decide what the subjects in experiment are, and what should be the tasks regarding the subjects. To ensure the validity of experimental results it is important to select the subject form the most appropriate population as hypothesis is applies to the whole populations by testing it on the selective sample of population. Experiment can be performed in different ways such as Randomized Experiment and Quasi-Experiment depending upon the type of research.

In randomized experiment which is also called true experiment researcher manipulates the independent variable to observe its effect on dependent variables by randomly assign the participants to groups in order to control the external factors from influencing the result.

Quasi experiment is used when normal experiment cannot be performed. In it, participants are not assigned randomly rather participant are pre-assigned to the groups based on some criteria such as gender, age software house etc. For example, if Quasi experiment is performed in an organization, then there might be constraints on working tasks of employees, that which employee will work on which specific task.

Experiment provides an inductive support for hypothesis, and its main concern is testing theories and proving hypothesis. If an experiment skips any single instance of an event which is predicted in theory or hypothesis, then theory or hypothesis might be rejected[65] [66] .

3.3.3 Card Sorting

Card sorting is a user-oriented activity, used to visualize the depth of customer's information. In it customers are provided with the set of cards, to organize and label them according to their understanding [67]. Each card holds some information. It is most reliable, inexpensive and simple method to gather users input for a comprehensive structure of any system. Two primary methods are used for card sorting open card sorting and close card sorting method.

i. **Open Card Sorting:** In this method participants are assigned the cards and are allowed to sort the cards according to their own understanding into different groups without any or very little restrictions and then describe the criteria for each group. This method is most appropriate when input to information is new or unclear and ambiguous.

ii. **Closed Card Sorting:** In this method participants are not allowed to sort the card according to their own understanding, rather to sort them out according to pre-defined criteria and to place the cards in already established groups or to add new cards into already existing sorted cards[68][69].

3.3.4 Action Research

Action research is conducted when researcher wants to solve some real-world problems and to study the problem-solving experience at the same time. Aim of an action research is to negotiate the studied situation, for improving the situation. It focused on combining theory and practical in a way that it provides practical solution to the customer while trying to acquire new theoretical knowledge at the same time. Action research can be defined as iterative approach where researchers and practitioners work together to identify the problem, solve the identified problem and to acquire new knowledge by solving the problem. Preliminary condition for an action research is that problem owner is willing to collaborate with researcher to identify the problem and to solve it. In action research problem owner becomes the research collaborator, and in some cases researcher and the problem owner are same person [65] [66].

3.4 Sampling

It is not possible for researcher to collect data from whole population to answer the research question, so there is a need to select the sample from the population that represent the whole population. Researcher needs to go through different stagers while conducting sampling. These stages are:

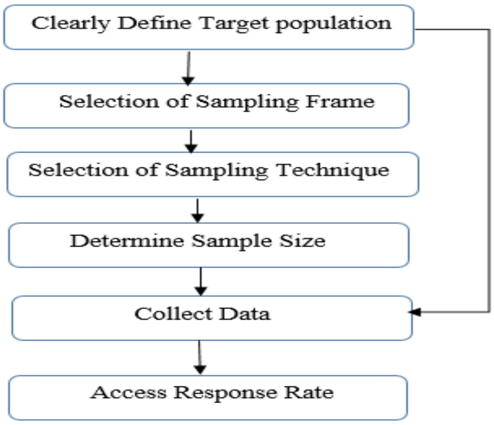
Clearly Define Target population
Selection of Sampling Frame
Selection of Sampling Technique
Determine Sample Size
Collect Data
Access Response Rate

FIGURE 3.1: Steps of Sampling Process

Stage 1: Clearly Define Target Population

In this first stage, researchers clearly define the population they want to target. In general population is the total number of people related to specific area or field.

Stage 2: Selection of Sampling Frame

Sample is generally the representative of whole population. Sampling frame is a list of real cases from which the sample is selected.

Stage 3: Selection of Sampling Technique

Sampling is a process of selecting subset from the selected sampling frame or population, to draw conclusion about population or to generalize in relative to existing theory. Sampling techniques are generally divided into two broader types i.e., probability sampling and non-probability sampling. Different techniques are associated with these two broader types.

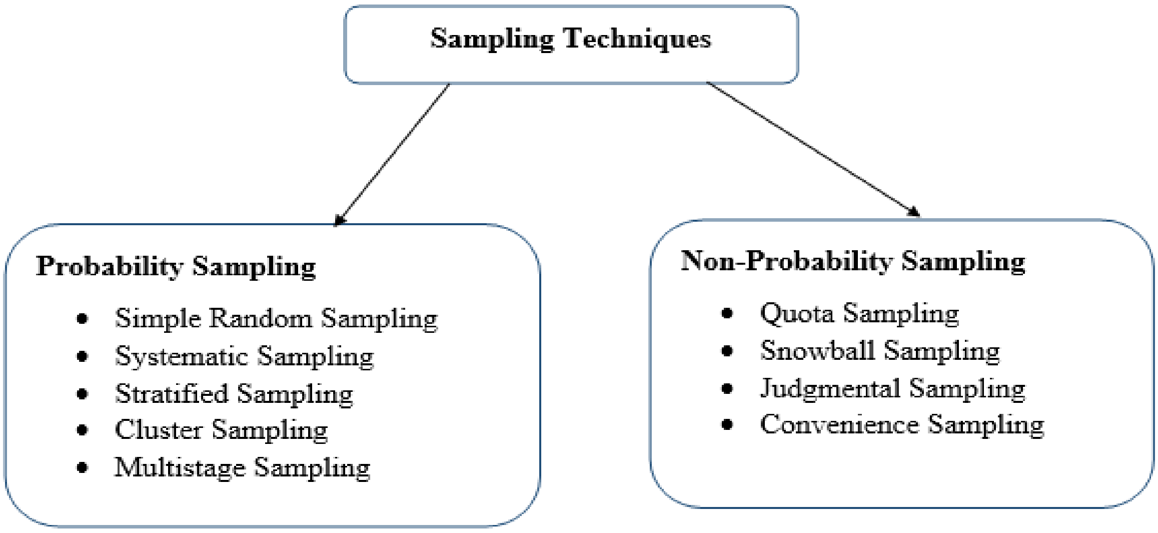

Figure 3.2: Sampling Techniques

3.4.1 Probability Sampling

Probability sampling is also known as Random sampling and in it, item has equal chances to be a part of sample. It is almost free from biasness but, may represent the most expensive sample in term of energy and time for a particular level of sampling blunder (Brown, 1947).

i. **Simple Random Sampling:** In this technique every item in population has equal chances to be a part of sample. For this, a complete frame of population is needed, and in some cases, it might be expensive if the data is collected by conducting personal interviews of the respondents who are geographically distributed.

ii. **Systematic Sampling:** This technique is simplest in nature where sample is selected based on some criteria or pattern i.e., select a random start and then pick every 5th sample from n number of population.

iii. **Stratified Sampling:** In this type of sampling, whole population is divided into different subgroups and then the sample is selected randomly form every subgroup. Subgroup may be created on the basis of some occupations, company size or gender. It is used when there is a great variation among population and its purpose is to ensure that every subgroup in the population represented equally.

iv. **Cluster Sampling:** In this sampling whole population is divided into small clusters or groups and then pick sample randomly from each cluster. It is beneficial for the researchers whose subjects are geographically distributed as it saves cost and time. Select the cluster for sampling the frame, such as cluster from some specific organization or from some geographical region, then assign the number to each cluster and pick up sample from each randomly.

v. **Multistage Sampling:** Purpose of this technique is to narrow the sample from a wider sample to make sample more practical, especially when either the complete list of samples does not available or not useable. It usually uses a combination of stratified or cluster sampling and random sampling. It is used when cost and time needs to be minimized.

3.4.2 Non-Probability Sampling:

Non-Probability sampling is usually associated with qualitative research and case study research design. Case studies are intended to study a real-life scenario by using small samples rather than making a statistical inference on the basis of large population. Sample in this study is not random or the representative of any population, rather a clear justification is needed for the selection of any individual sample for specific case.

i. **Quota Sampling**: In this technique, sample is select on the basis of some predefined criteria, so all the samples will have same characteristics of distribution.

ii. **Snowball Sampling:** In this non-random sampling technique, few cases are used that helps and encourage other cases to be a part of study. This is most suitable when the nature of population is closed and is difficult to access the potential participant among population.

iii. **Judgmental Sampling:** Judgmental Sampling is also known as purposive sampling. In it any person or case is selected consciously to get some important information's that cannot be obtained otherwise. It is convenient to use as it saves cost and time and is most suitable for exploratory studies.

iv. **Convenience Sampling:** In this technique participants are selected as sample because they are easily available and ready to answer. This is considered as favored sampling, as it is cheap and an easy choice comparatively to other sampling techniques.

Stage 4: Determination of Sample Size

To avoid biasness and errors in sampling, and to generalize sample to population it is very important to have adequate size of samples. Here, adequate sample does not mean the ratio of sampled population, but the authentic size of the selected sample related to the complexity of population. Greater the sample size lowers the chances of errors and biasness in findings. Numerous statistical formulas are available to determine the sample size. But the formula used for the determination of sample size of the population does not have any practically effect on how well the sample size define the population as according to Fowler, it is very infrequent to consider on the part of population when deciding the size of sample.

Sample size demonstrate the number of responses obtained rather than the number of distributed questionnaire as the number of distributed questionnaire is usually more than obtained responses, to compensate the unresponsive questionnaire.

Stage 5: Data Collection

Data is collected after deciding the targeted population, sampling frame, techniques to be used for sampling and the selected size of sample.

Stage 6: Assess Response Rate

The number of cases who are agreed to be a part of study and to respond response rate. These cases are selected from real sample. It is very rare for the researchers to achieve 100% response rate due to many reasons, such as incapable to respond, unwillingness to respond, not eligible to respond or respondents might be available, but researchers are unable to reach or contact them. In addition, response rate is very important as every nonresponse is responsible for the biasness in concluding sample, so clearly defined sample, use of right sampling technique and using large samples might help to reduce the biasness in sample[70].

3.5 Qualitative Research

Quantitative research methodology is effective either when researcher investigates new field of study or aiming to discover and formulate the important issues. Different quantitative methods or instruments are available that are used for in-depth and detailed understating of data. [71] Most commonly used methods for qualitative research are individual interviews, laddering focus groups, observation, and case study.

3.6 Instruments of Qualitative Research

Following instruments are used to conduct qualitative research:

3.6.1 Individual Interview

Individual interview is one of the most popular instruments used in qualitative method of research. Individual interview can be designed in three different ways depends upon the researchers and the type of research. Interviews can be structured, semi-structured and un-structured[72]. Interview can be conducted either In-person or through telephone.

i. **Structured Interview:** In this type of interview, researcher provide all the questions that will be asked during interview to the participants. Participants prepare the short answer to the provided questions. Interview is scheduled for a certain time and no additional question or the questions other than the already provided questions could be asked. This instrument limits in a way that only rating questions or close ended question with answers of yes or no can be asked that does not provide in-depth detail of the data so, limited information can be gathered using this instrument.

ii. **Semi-Structured Interview:** This type of interview is also known as focused interview. In it limited structure is found and most of the questions are open ended where participants can answer the asked questions in detail depends upon the research topic and the areas that need to be covered. Questions asked in it are broad, and participants or interviewee are encouraged to answer the questions. This method of interview provides an opportunity to have a detailed discussion among interviewer and interviewee by staying within the topic boundaries. In it interviewer or researcher is allowed to direct the interview depends upon the quality of answers from interviewee[72].

iii. **Un-structured Interview:** This is also called in-depth interview. In it questions are not pre-planned or prepared unlike other two methods of interview and in it, researcher or interviewer prepare himself to discuss different topics with interviewee rather than asking the questions. The aim of this method is to gather in-detailed information on the topics needed. It enables the detailed discussion among interviewee and interviewer on desired topics that leads to collect the as detailed data as possible. As in it there is no pre-planned or prepared questions, so interviewer needs to be prepared for the discussion depending upon the already available information from previous interviews.

3.6.2 Laddering

Laddering is an interview technique, used for conducting semi-structured interviews. In it, stakeholders give answer to different simple questions in clear way. The questions that will be asked, organized into a hierarchy, to show the sequence in which question will be asked. This method enables the researcher to closely contact the stakeholders to know about priorities, and then organize them in a hierarchal manner so it is easy to understand. If requirements kept adding and removing constantly in laddering then it is very hard to maintain it [67].

3.6.3 Focus Group

Focus groups are frequently used method in action research. It is basically a form of group interview with 4 to 10 participants. Focus group is preferred over individual interviews in some cases where it is considered that the quality of information gathered through focus group is much better and effective as compared to individual interviews. This method is very effective to use in the beginning of the research where researcher needs to narrow down or explore the specific topic. Moreover, it is most suitable when the collective decision is required of some matter. It is also preferable when resources are limited such as cost, which is measured per hour, so researcher can get 4 to 10 opinions at a time rather than single opinion unlike individual interview. Multiple focus groups can be conducted in order to gather more information or to get conformity of something. It limits in a way that only limited number of question can be asked due to time constraints and it is very hard to fairly divide the time with each participant and participants have a very short time to share their opinions[62][72].

3.6.4 Observation

Observation is another method of data collection used in qualitative research. It is suitable when respondents are unable to give the verbal report of their feelings due to any reasons. Moreover, it is independent to the willingness of the respondents of respond. Observation can be structured, unstructured, participant, or non-participant observation.

i. **Structured Observation:** In structured observation the environment where investigator or researcher is observing everybody knows that someone is observing their work. This method limit in a way that when participants are aware of the fact that they are being observed so, there is a possibility of changing behavior of the participants. Moreover, they may act unnatural in front of observer.

ii. **Unstructured Observation:** In this method no one in the working environment knows that they are being observed by someone.

iii. **Participant Observation:** In this kind of observation observer is allowed to observer by staying within the situation and by taking part in the group which is being observed. Observer can interact with all the participants in the group to learn about their behavior and activities as a group member. By closely observing the groups as a member of group and by studying the natural behavior and real characters observer can get better understanding of the situations.

iv. **Non-participant Observation:** In this observation observer observe the situation from a distance and is not allowed to participate in any kind of group activity. It is very hard to get the proper understanding of the matter without being involve into it so, sometimes combination of both participants and non-participants methods are used to get the better understanding of the matter.

Observation limit in a way that it is dependent to the situation and the understanding and judgmental ability of the observer, so observer may skip any critical or important point due to any reason like might get distracted due to any factor or forget to write any important point while writing what he has observed. There are different available techniques to collect data through observation that can overcome the limitation of this technique such as description writing, which is the initial step in observation, another one is video recording from where observer can reassure the data later. Researchers can use any or all the techniques collectively to gather reliable information [3].

3.6.5 Case Study

Case study was originally developed and used for exploratory research, but it can be used for descriptive research depending upon the situation of phenomena of the study. It might be used as explanatory studies for describing the pre or post events of research.

Case studies are conducted in real world environment so, are highly realistic in nature. Case study is conducted when researcher wants to dig deeper into something to get detailed data, so in it any required case is consider as a scenario and from that scenario any instance get picked that needs to understand in detail. Case studies are flexible in nature so in it all qualitative method can be used to understand the picked instance or whole scenario so any method can be choose depending upon the scenario or the instance that is picked. For example, if researcher is unable understand the situation that is told by any other employee so researcher may join the team as an observer to get the better understanding of the situations. Or if researcher wants to discuss or ask something from higher administration, he may conduct an interview of higher administration of the organization.

For conduction of case study five major steps are involved:

i. Design a case study: plan a case study by defining its objectives.
ii. Data Collection: Define all the procedures and protocols that are used for data collection.
iii. Evidence collection: Execute the collected data with studied cases.
iv. Analysis of data.
v. Report writing.

The mentioned steps are almost similar for any other type of study such as empirical study but, as case study is flexible so data can be collected in number of iterations. Like if researcher has collected data and found data insufficient, then more data collection can be planned. Case study also limit in term of flexibility in a way that objective of study should must be clear before starting case study because if objective changes in later iteration then it will be a new case study rather than an addition or change to the existing case study [73].

3.7 Mixed Methodology

Mixed method research is a combination of more than one research methods, which involves quantitative and qualitative methods for conducting research. It is a contrast of quantitative and qualitative methods that is used either sequentially or concurrently. Most familiar strategies of using mixed method approach are Sequential explanatory, Sequential exploratory and Concurrent triangulation method described by Creswell.

Sequential explanatory method is used when an explanation of quantitative result is required, so qualitative method is used to explain the situation by interpreting the quantitative results. It is used when an unexpected result arises in quantitative methods.

Sequential exploratory method is used when testing is required for an emerging theory found in qualitative study or when the interpretation of qualitative data into quantitative data is needed.

Concurrent triangulation is widely used method in mixed method research. In it, different methods work parallel as an independent of each other to confirm and validate the findings. By simultaneous collection of data from both types, analysis can be used to explore the emerging results of other methods. This method is challenging in a way that analyses of different methods might be tough for researcher, moreover if any contradiction is found in results it could be hard to resolve and any follow up study could be required to resolve the contradictory issues [65] [74].

3.7.1 Three-Dimensional Mixed Method Research Design Topology

As a result of the content analysis of the various mixed research plans available, it was conceived that these plans can be interpreted as a function of the following three aspects such as level of mixing, time orientation and emphasis of approaches. Level of mixing demonstrate that either the methodology is partially mixed or fully mixed, similarly time orientation demonstrate that either both quantitative and qualitative methods are being used at the same time in concurrent way or one method needs to be completed for starting another method in a sequence and in last level which is, emphasis of approaches, demonstrate the significance of each approach whether both methods have equal priority for answering the research question or any method has higher priority comparatively with other method.

Basic difference between fully mixed and partially mixed design is that in fully mixed design quantitative and qualitative methods are mixed within one or more phases of research or through the research process, while in partially mixed research design quantitative and qualitative methods are not mixed within or through research process rather in this both quantitative and qualitative method research is conducted either sequentially or concurrently as a whole before being mixed for data analysis and interpretation[75].

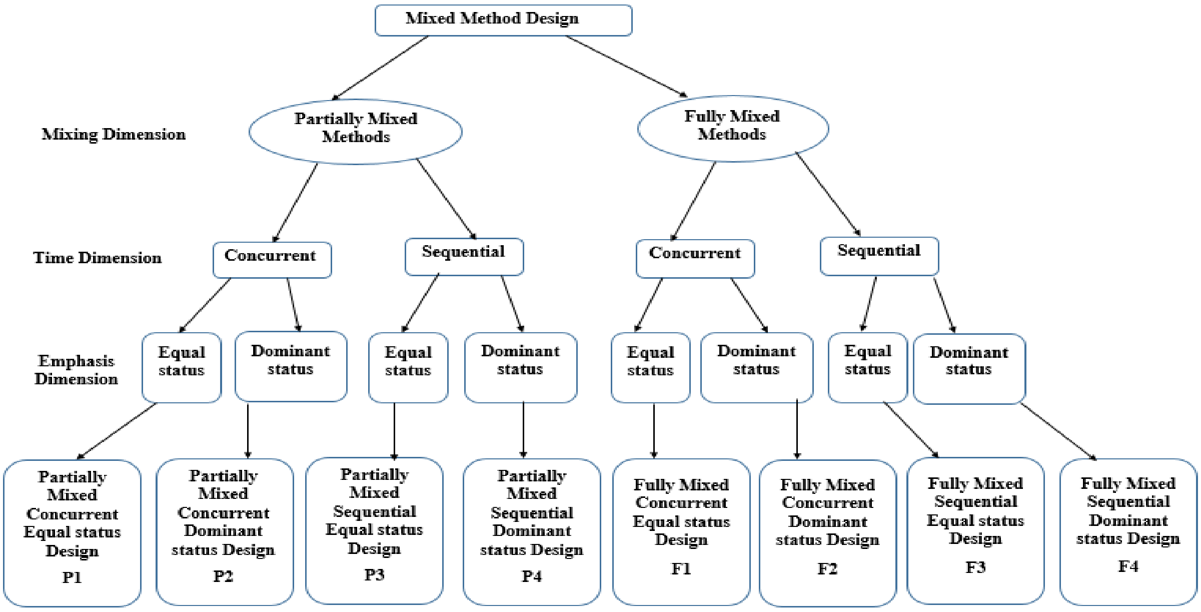
Mixed Method Design
Mixing Dimension
Partially Mixed Methods
Fully Mixed Methods
Time Dimension
Concurrent
Sequential
Concurrent
Sequential
Emphasis Dimension
Equal status
Dominant status
Equal status
Dominant status
Equal status
Dominant status
Equal status
Dominant status
Partially Mixed Concurrent Equal status Design P1
Partially Mixed Concurrent Dominant status Design P2
Partially Mixed Sequential Equal status Design P3
Partially Mixed Sequential Dominant status Design P4
Fully Mixed Concurrent Equal status Design F1
Fully Mixed Concurrent Dominant status Design F2
Fully Mixed Sequential Equal status Design F3
Fully Mixed Sequential Dominant status Design F4

Figure 3.3: Topology of Mixed Method Research Design

Total 8 types of mixed method design are derived by crossing all three dimensions such as partially vs. fully mixed, concurrent vs. sequential and equal vs. dominant status. The types that are derived by crossing dimensions are:

i. **Partially Mixed Concurrent Equal Status Design(P1):** In this design study is conducted by partially mixing the quantitative and qualitative methods. Both the methods work concurrently and having equal priority and role in answering the research questions.

ii. **Partially Mixed Concurrent Dominant Status Design(P2):** This method involves partial mixing of quantitative and qualitative methods for conducting study. In it both methods work concurrently but one of the methods has high priority than other method for answering the research question, so method with high priority is dominant over other method.

iii. **Partially Mixed Sequential Equal Status Design(P3):** In this method study is conducted in two phases. Both quantitative and qualitative methods have equal priority for answering the research questions, but one method needs to be completed for starting other method as it works sequentially i.e., in first phase qualitative method is used to collect data and in second phase data is analyzed by using quantitative methods.

iv. **Partially Mixed Sequential Dominant Status Design(P4):** In this design methods are partially mixed and study in conducted in two phases where one of the methods from quantitative and qualitative method is dominant over other as the priority of one method is high to answer the research question comparatively to other method. As this design use sequential study method so one of the methods from quantitative and qualitative method is completed first and then second method is adopted in second phase.

v. **Fully Mixed Concurrent Equal Status Design(F1):** In this method quantitative and qualitative methods are fully mixed for conducting the study. In single research quantitative and qualitative methods are concurrently mixed either in one or more steps or among all of these four components: objective of research, data types and functions, analysis type and conclusion type of data. In it both methods have equal priority for answering the research question.

vi. **Fully Mixed Concurrent Dominant Status Design(F2):** In this design for a single research quantitative and qualitative methods are concurrently mixed either in one or more steps or among all of these four components: objective of research, data types and functions, analysis type and conclusion type of data. Unlike F1, in this design one of the methods among quantitative and qualitative methods have higher priority for answering the research question.

vii. **Fully Mixed Sequential Equal Status Design(F3):** In this design for a single research quantitative and qualitative methods with equal priority to answer the question are sequentially mixed either in one or more steps or among all of these four components: objective of research, data types and functions, analysis type and conclusion type of data.

viii. **Fully Mixed Sequential Dominant Status Design(F3):** This design is similar to F3 except the priority of quantitative and qualitative methods. In this design one of the method among quantitative and qualitative methods have higher priority for answering the research question, so the method with higher priority is dominant over other method[75].

3.8 Research Context and Justification

This section discussed the methods, used to conduct the research. Partially Mixed Sequential Dominant Status Design (P4), has used to conduct the research. We conducted industrial surveys as quantitative study to validate the challenges identified through SLR and as quantitative study, we conducted expert review to validate the framework proposed in our study to mitigate the identified challenges.

First, we conducted SLR, to review the existing literature related to distributed scrum. The reason of choosing this method was that it provides a broad review and critical analysis of all related existing research on and around the area of distributed Scrum. It is a predefined strategy that helps in conducting reasonable literature review in an organized way. The primary objective of selecting SLR was, to identify the challenges confronted by distributed scrum teams along the mitigation strategies adopted by teams to treat the confronted challenges. Besides this research methodology, an industrial survey conducted and selected as second research methodology in order to validate the challenges identified from literature review. Moreover, we conducted an expert review to evaluate the effectiveness of framework we proposed in this research.

3.8.1 Systematic Literature Review (SLR)

Systematic literature review selected as first method to conduct research. It helped us to discover all the possible key challenges, confronted by distributed scrum teams. For performing SLR we selected the best standard guidelines by Kitchenham [76], as it is the most comprehensive guideline for conducting SLR in field of software engineering. The selected SLR standard guideline covered three parts: Review planning, Review conduction and Results reporting. Figure 3.4 shows the overview of the SLR steps.

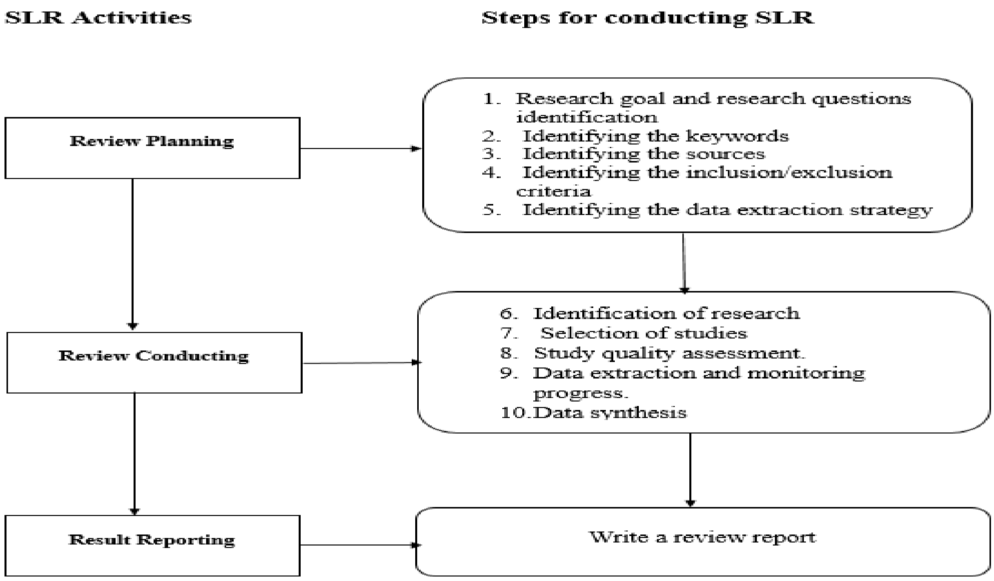
SLR Activities
Steps for conducting SLR
Review Planning
1. Research goal and research questions identification
2. Identifying the keywords
3. Identifying the sources
4. Identifying the inclusion/exclusion criteria
5. Identifying the data extraction strategy
Review Conducting
6. Identification of research
7. Selection of studies
8. Study quality assessment.
9. Data extraction and monitoring progress.
10.Data synthesis
Result Reporting
Write a review report

Figure 3.4: Overview of SLR Stages

The data collection phase of SLR started, after describing the research objectives. In planning phase of SLR, first task was to identify the research objective and research questions, on which the whole research is based, and that is already done and defined in chapter 1. Next task was to develop a review protocol, which included: the keywords used in search string to extract the related papers, sources from where the related papers got selected, criteria to include or exclude the papers and lastly strategies used to extract data.

In second step of SLR which is review conducting, we used our chosen search strategy to find the related research papers, and then select the papers based on defined inclusion/exclusion criteria. We reviewed the quality to determine the quality of the selected papers. After getting basic papers, we used snowball sampling technique to get as much relevant papers as possible, and it helped to fulfil the deficiency of excluded or irrelevant papers. Snowball sampling and quality assessment performed until no new paper was found, or the found papers were out of research scope. After that, we extracted and synthesized data from selected papers by reading them carefully.

The goal of this SLR was identify the confronted key challenges while executing distributed Scrum (RQ1). Secondly, in order to handle the identified challenges, some mitigation strategies were also identified through SLR (RQ2). all steps involved in SLR are discussed in detail.

Step 1: Search Strategy

It involves 4 basic steps including identification of keywords, selection of sources, generating query strings and conducting research.

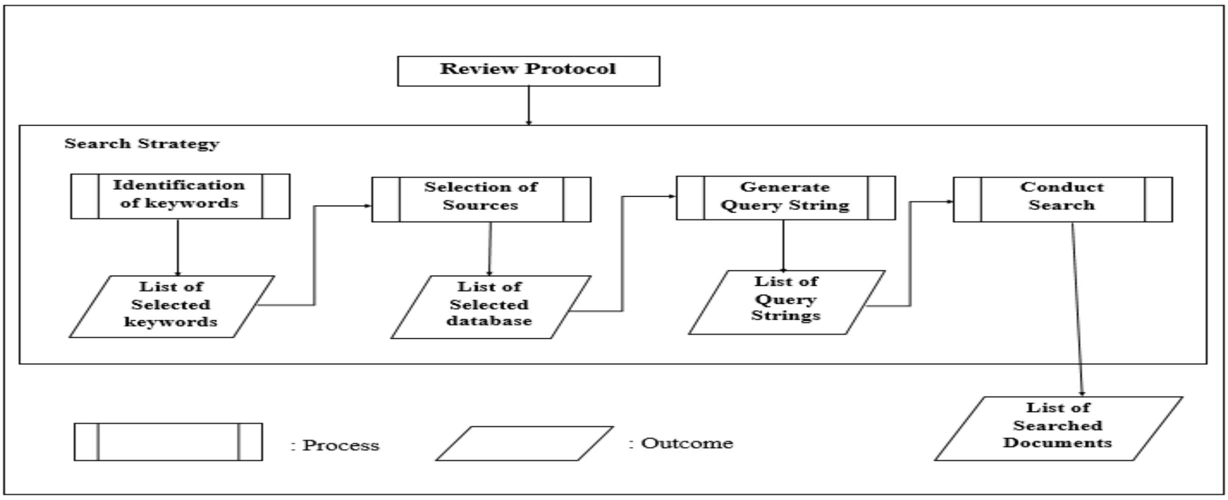

Figure 3.5: Overview of Search Strategy

Search process started with the identification of keywords, which helped us to search and get our desired relevant papers. We selected three words relevant to our research topic which is challenges, distributed and scrum. Then combined these keywords and generated as many query strings as possible to get as many papers as possible related to our topic. We used the principal that the selected keywords include all its synonyms so here, global is used as a synonym of distributive keyword and issues, problems and risks are used as synonym of challenges, respectively. Moreover, after reading some papers, we realized that although the theme of some studies is to find out the challenges and issues of distributive or global software development, their research is based on agile methods referred to scrum or on scrum frameworks, so we added the broader term, agile in addition to scrum. Keywords were selected based on three basic concepts: challenges, distributed software development and use of scrum methods as shown is table below.

Table 3.1: Search Keywords

Challenges	Issues, Problems, Risks, Challenges
Mitigation	Resolution, Solution, Reduction, Mitigation
Distributive Software Development	Global Software Development, Distributive Software Development, Geographically Distributive Development, Distributive Software engineering, Global Software Engineering.
Use of Scrum Methods	Agile, Agile Methods, Scrum, Scrum Methods

Query string is created by using identified keywords and Boolean connectors OR, AND. OR connector is used while writing similar words or group of words with same meaning AND is used as connector where different words are used.

Five different digital database sources were selected and used for getting primary literature related to our topic. The selected sources are focused on software engineering and computer science papers and collect peer-reviewed articles. Information regarding query string and database sources is mentioned in table below:

Table 3.2: Query String and Selected Databases

Query String	("Issues" OR "Problems" OR "Risks" OR "Challenges") AND ("Resolution" OR "Solutions" OR "Reduction" OR "Mitigation") AND ("Global Software Development" OR "Distributive Software Development" OR "Geographically Distributive Development" OR "Distributive Software engineering" OR "Global Software Engineering") AND ("Agile" OR "Agile Methods" OR "Scrum" OR "Scrum Methods")
Digital Database Sources	i. IEEE Xplore (https://ieeexplore.ieee.org/Xplore/home.jsp) ii. Springer (https://www.springer.com/in) iii. ACM Digital Library (https://dl.acm.org/) iv. Wiley Online Library(https://onlinelibrary.wiley.com/) v. Elsevier (https://www.elsevier.com/)

Step 2: Inclusion/Exclusion Criteria

Huge number of papers were found from our defined query strings, which aimed to get all the papers that could be valuable for research. An inclusion/exclusion criteria has been used to screen the acquired papers, and to judge which papers are the most relevant to our research topic and which papers are irrelevant and should be excluded [77]**.**

Table 3.3: Inclusion/Exclusion Criteria

Inclusion Criteria	**Exclusion Criteria**
i. Paper necessarily be book chapter, journal, conference, or magazine.	**i.** Paper other than book chapter, journal, conference, or magazine
ii. Paper must be in English.	**ii.** Paper other than English language.
iii. Full content must be shown.	**iii.** Full content is not shown.
iv. Paper discuss the challenges associated with distributed scrum.	**iv.** Paper does not discuss the challenges associated with distributed scrum
v. Paper must be a peer reviewed.	**v.** Paper does not a peer-reviewed
vi. Paper published in domain of distributed scrum from year 2005- 2020.	**vi.** Duplicate papers or papers published before 2005.

Step 3: Quality Assessment

After extracting the papers by applying inclusion/exclusion criteria, there was need to assess the quality of extracted papers. Quality assessment of primarily studies is important as it helps to limit the biasness while conduction systematic review, to get understanding of the possible comparisons and to get the interpretations of the results [78]. It provides further exhaustive inclusion / exclusion criteria. The main purpose behind the quality assessment is to make sure that, the extracted data is relevant to our research and is unbiased. Moreover, quality assessment of primary papers can help us to understand the importance of individual studies while compiling results.

Different quality assessment tools, checklists and guidelines are available in software engineering, for assessing the quality of each primary study [79].

We used a quality checklist to access the quality of our primary studies. The question included in checklist are:

i. Is there a clear statement of the aim of research?
ii. Is there sufficient explanation of the context in which the research was conducted?
iii. Is there a clear statement of findings?[78]

3.8.2 Snowball Sampling

Snowball sampling is an extremely useful method for extending the SLR studies. The first step in snowball sampling is to identify the set of preliminary papers that will be used for snowball sampling based on the inclusion/exclusion criteria. It is very important to make sure that the paper used for snowballing will be used in the final analysis, as only papers that are found through the included papers would be used in analysis, and if it realized later that paper used for snowball sampling should not be included, the process will have to be reversed and papers that are incorrectly included have to be removed[80]**.**

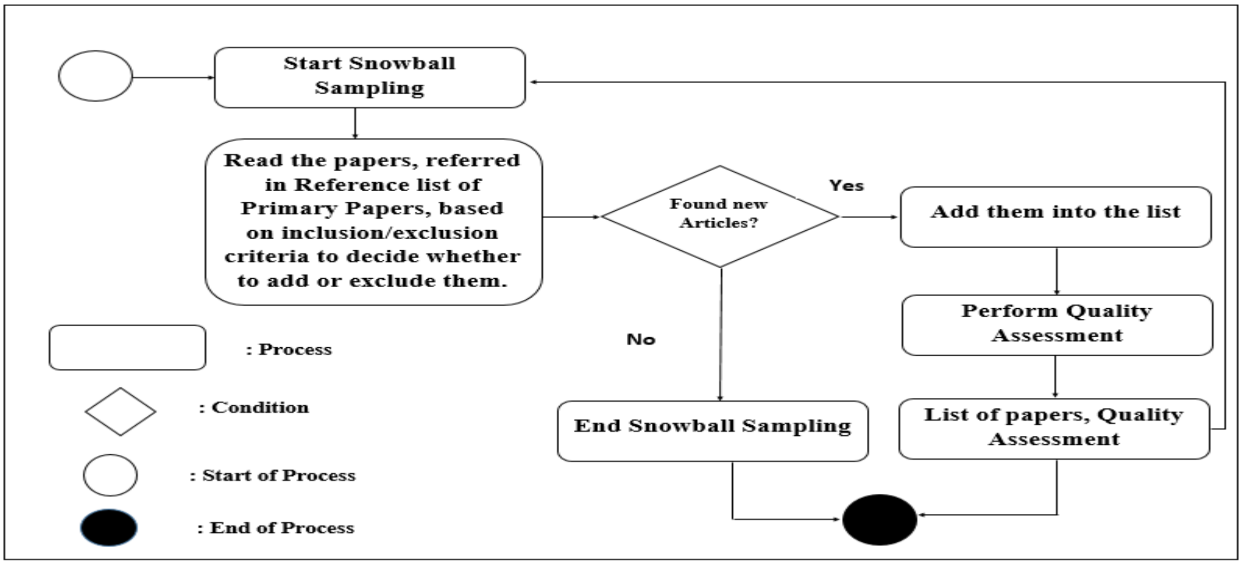
Start Snowball Sampling
Read the papers, referred in Reference list of Primary Papers, based on inclusion/exclusion criteria to decide whether to add or exclude them.
Found new Articles?
Yes
Add them into the list
Perform Quality Assessment
List of papers, Quality Assessment
No
End Snowball Sampling
: Process
: Condition
: Start of Process
: End of Process

Figure 3.6: Overview of Snowball Sampling Process

Snowball sampling began after accessing the papers from defined selection criteria. We read the papers referred in the list of primarily selected papers and based on inclusion/exclusion criteria, papers got selected. After getting the list of new papers through snowball sampling, their quality assessed, and kept performing snowball sampling until no new paper was found. Snowball sampling ended after getting all the articles that would be use in final analysis.

Step 4: Data Extraction and Synthesis

After quality assessment we got our required papers, now we were needed to identify the most common issues encountered by team during distributed scrum development. So, it began with reading all the selected articles carefully and all the challenges discussed in each paper were documented. After documenting all the discussed challenges, we counted how frequently each challenge has been mentioned in papers and by comparing the frequency of every challenge we got the list of most common challenges encountered during distributed scrum development to fulfil our (RObj1) and to answer our (RQ1).

So, through data extraction and synthesis process we got the list of most common challenges encountered during distributed scrum development which helped us to indicate the direction and scope of our research.

3.8.3 Survey

The second selected methodology as mentioned above was an industrial survey. The main purpose of selecting this method as our research methodology was to validate the challenges identified from literature review. For this purpose, we followed the Kasunic guidelines [64] which is published by Software Engineering Institute (SEI). His work is followed because it is the most general reference guideline that is world widely used for conducting an effective survey in software engineering field. Figure 3.7 shows the steps for survey conduction.

3.8.4 Objective of Conducting Survey: The objective for survey conduction was:

Objective 1: To validate the challenges identified through SLR by distributed scrum practitioners.

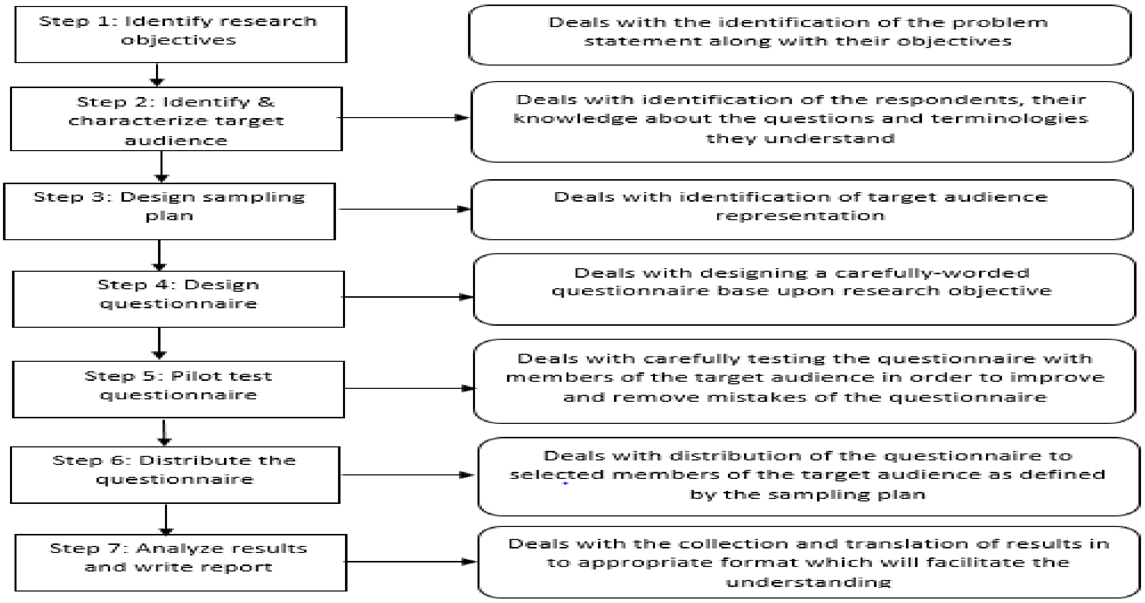
Step 1: Identify research objectives
Deals with the identification of the problem statement along with their objectives
Step 2: Identify & characterize target audience
Deals with identification of the respondents, their knowledge about the questions and terminologies they understand
Step 3: Design sampling plan
Deals with identification of target audience representation
Step 4: Design questionnaire
Deals with designing a carefully-worded questionnaire base upon research objective
Step 5: Pilot test questionnaire
Deals with carefully testing the questionnaire with members of the target audience in order to improve and remove mistakes of the questionnaire
Step 6: Distribute the questionnaire
Deals with distribution of the questionnaire to selected members of the target audience as defined by the sampling plan
Step 7: Analyze results and write report
Deals with the collection and translation of results in to appropriate format which will facilitate the understanding

Figure 3.7: Phases of Survey Conduction

Step 1: Identify Research Objective:

Clear identification of research objective is very important before conduction of survey, because it helps to determine the scope of survey by determining who should be the respondents and what questions should be asked from them [64].

Step 2: Identify the Target Audience

In this step we identified our targeted population that has been used for our survey study. In survey, population is the whole audience we want to draw conclusion about. When the population is identified, it becomes the target audience of research. Audience is selected based on research perspective depending upon the research objectives and by determining who can provide the best information related to our research objective. After identification of targeted audience, we select samples for research. Samples are the subset of population that represent the whole population. In survey, sample is studied and results are generalized throughout the population [64].

As we were conducting an industrial survey, so our targeted audience was agile and scrum practitioners who has been working in industry and doing development using agile scrum methods.

Step 3: Designing the Sampling Plan

In this step we determined how individuals will be selected to take part in survey and what will be the size of sample. Sample size can be determined by using different formulas. Determination of appropriate size of sample is a function of desired precision, desired confidence level and population size [64].

i. **Precision:** Precision measures how close the characteristics are, of actual and estimated population. Level of precision depends upon acceptance level of risks that how much risks are tolerable while making decision. If we want to increase precision, sample size will also increase [64].

ii. **Confidence Interval:** Confidence interval is a range of values, we are sure about that our true values will be lie within this specific range. Standard deviation is used to calculate the confidence interval of samples or population [64]

iii. **Confidence level:** Confidence level is that how much confident we are about our selected sample. Suppose we select 100 samples as representative of our targeted population, then how much we are sure that how many samples from these 100 samples are free of risks. If we say 95 out of 100 samples are risk free than it means that we are 95% sure about our selected samples. To find out the confidence level, confidence interval is required. Standard Normal distribution and Central Limit Theorem is used to calculate confidence level by using z-values [64].

iv. **Population Size:** Sample size calculation is also affected by the size of population when population size is small. Different formulas are used to calculated sample size depending upon the size of population whether it is large or small [64].

Step 4: Design and Write the Questionnaire Overview

In this step we identified what are the questions that will be asked, what will be the type and format of questions and in which sequence questions will be asked.

3.8.5 Type of Questions

Four main types of questions are available that could be asked from respondents depending upon the type of research. These types are attributes, attitudes, beliefs and behaviors [64].

In attributes type of question, questions are typically asked about personal or demographic information such as occupation, experience, age etc. [64]

In attitude type of question, questions are asked about how people feel about some certain things to find out whether their feelings about some certain things are positive or negative [64].

In belief type of questions, questions are asked about people belief of something. This type of question is more focused and can be referred as opinion question, in which question are asked to know about the belief of people about some certain thing and try to get their opinion on it [64].

In behavior type of question respondents are asked about their behavior that what they have done in past and what they are presently doing. In concrete terms it is about their belief on their behavior [64].

After deciding the type of questions that will be used in our research work, we decided the response format of the question that which type of question will be asked either open ended or close ended or hybrid question.

i. **Likert Scale:** Likert scale is the most used instrument in survey for assessing the respondent's attitude about certain things. Single choice, close ended questions are used for Likert scale assessment. It helps to acquire more granular information or perspective of respondents about certain things than a simple yes/no question. Usually five to seven items are used in Likert scale [81]. In our research we have used five item Likert scale to assess the respondent's opinion.

For our research we have used attribute and attitude type of close ended question and used likert scale to ask the question. These types of questions are used as these questions are the best suits to obtain our (RObj1) and to answer our (RQ1). Questions asked in survey are mentioned in table 3.3 and 3.4 below.

Step 5: Design Test Questionnaire

After selecting the type of questions used in this research, we prepared a questionnaire and shared with supervisor for pilot tests, to remove mistakes and to make improvements in questionnaire to ensure that the asked questions will be understandable by all in true context.

Step 6: Distribute the Questionnaire

After finalizing the questionnaire by conducting pilot study, we distributed the questionnaires in distributed scrum practitioners. We conducted Online survey and the participants of this survey were selected by applying different filter while searching for the appropriate participants on linked In. After identification of appropriate participants, we shared the questionnaire with them to get to know their opinions about identified challenges. Questions asked in survey are mentioned in table 4 below.

Table 3.4: Attribute Questions Asked in Survey

Demographic Questions	
Q.1 Which of the following best describe your role in organization?	○ Upper Management ○ Middle Management ○ Team Member
Q.2 Size of your team?	○ <15 ○ 16-25 ○ 26-35 ○ 36-45 ○ 45>
Q.3 Your experience in Distributed Scrum?	○ 0-3 Years ○ 4-7 Years ○ 8-10 Years ○ More than 10 years

Table 3.5: Attitude Questions Asked in Survey

Specific Question about Distributed Scrum					
(Strongly Agree = **SA**, Agree = **A**, Neutral = **N**, Diagree = **D**, Strongly Diasgree = **S DA**)					
	S A	**A**	**N**	**D**	**S DA**
Q.1 Would you agree that Geographical distance among teams leads to various challenges?	○	○	○	○	○

Q.2 Would you agree that Socio-cultural difference among teams causes different challenges?	○	○	○	○	○
Q.3 Would you agree that temporal difference among teams causes different challenges?	○	○	○	○	○
Q.4 Would you agree that Communication & Coordination is one of the biggest challenges in distributed scrum?	○	○	○	○	○
Q.5 Would you agree that lack of collaboration among distributed scrum teams causes significant issues while development?	○	○	○	○	○
Q.6 Would you agree that continuous integration management in distributed scrum is a challenging task?	○	○	○	○	○
Q.7 Would you agree that Configuration management in distributed scrum is challenging task?	○	○	○	○	○
Q.8 Would you agree that Risk management is a challenging task in distributed scrum development?	○	○	○	○	○
Q.9 Would you agree that Quality Assurance is a challenge while working in a distributed environment?	○	○	○	○	○
Q.10 Would you agree that lack of resistance to change causes many significant challenges in distributed scrum?	○	○	○	○	○
Q.11 Would you agree that for the success of software, clear understanding of software architecture among teams and stakeholders is very important?	○	○	○	○	○
(Always = **A**, Frequently = **F**, Sometimes = **ST**, Seldom = **S**, Never = **N**)					
	A	**F**	**ST**	**S**	**N**

Q.12 How often your team have faced software architectural understanding issues in distributed scrum?	○	○	○	○	○
Q.13 How often your team have faced knowledge sharing and management issues in distributed scrum?	○	○	○	○	○
Q.14 Proper project management is very important for the success of project. How often your team have faced poor project management issues in distributed scrum?	○	○	○	○	○
Q.15 Requirement engineering plays vital role for the development of desired product. Have your team ever faced requirement engineering issues in distributed scrum environment?	○	○	○	○	○
Q.16 Shared understating is very important for all the teams working on same project in distributed scrum. How often your team have faced shared understating issues?	○	○	○	○	○
Q.17 Trust among teams for sharing important information with each other is very important. Have your team ever faced trust issues?	○	○	○	○	○

3.9 Expert Review

Expert review is the third and last methodology used in our research. The main purpose of selecting this method as our research methodology is to validate the effectiveness of our framework proposed in this research. We conducted expert review by following Bilal Ayyub guidelines [82]. Figure 3.8 shows the steps followed to conduct expert review.

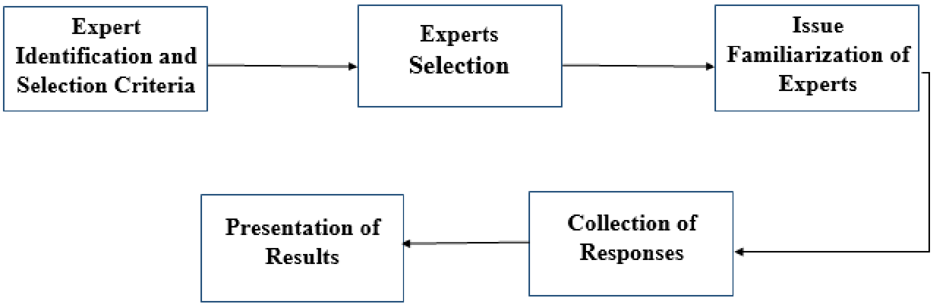
Expert Identification and Selection Criteria
Experts Selection
Issue Familiarization of Experts
Collection of Responses
Presentation of Results

Figure 3.8: Steps of Expert Review

Step 1: Expert Identification and Selection Criteria

To identify and select the experts for our expert review, we set the following criteria:

- Strong knowledge and expertise in distributed scrum.
- Familiarity with various issue related aspects.
- Willingness to effectively participate in expert review
- Available to give the required time and participate in expert reviews.

Step 2: Experts Selection

By following the defined criteria, 5 experts got selected with more than 5 years of experience in distributed scrum who were willing to effectively participate in our expert review and to share their opinion about our proposed framework.

Step 3: Issue Familiarization of Expert

When experts got selected, we shared our data with experts. The purpose behind the sharing of data was to ensure that all experts get common understanding of the issues. The data that was shared with experts included: identified challenges, framework we proposed, and the questions related to the framework, to evaluate the effectiveness of framework. Table 5 shows the question asked in expert review.

Table 3.6: Questions Asked in Expert Review

Question
1. Communication and coordination issues among teams could be resolved by maximizing synchronizing working hours of distributed teams. What is your opinion on this?
2. Providing high quality simulation tools to the teams and allowing them to communicate openly can resolve communication, coordination, and collaboration issues. What is your opinion on this?
3. Encouraging informal communication among teams can helps the teams to build trust upon each other? What is your opinion on this?
4. Pair programming can help the teams to build trust upon each other. Do you agree?
5. Integrating user stories with use cases can be helpful to mitigate requirement engineering issues. Do you agree?
6. Follow-up question with formal document of requirements can helps to ensure clear understanding of requirements and requirement priorities. Do you agree?
7. Carefully managed requirement changes and by ensuring clearly visible priorities can help to overcome software architectural issues? What is your opinion on this?
8. Project management issues can be mitigated by focusing on people management. What is your opinion on this?
9. Providing essential trainings to teams and by taking regular feedback from stakeholders about work can help to manage project more effectively. What is your opinion on this?
10. By introducing standard mechanism for knowledge sharing and management can help to reduced knowledge sharing and management issues. Do you agree?
11. Maintaining a management system to manage and maintain the knowledge shared with distributed teams can help to overcome knowledge sharing and management issues. What is your opinion on this?
12. Trainings on different cultures can help the teams to equally understand the knowledge shared within teams. Do you agree?

13. Using common tools for development by all distributed teams can help to mitigate shared understanding issues. Do you agree?
14. Unified backlog shared and accessible to all teams can help to overcome shared understanding issues?

3.10 Summary

In this chapter we have discussed available research methods in terms of quantitative and qualitative methods and mixed method research. Then the research methods used in this research are discussed in detail in terms of research context and justification. We have used Partially Mixed Sequential Dominant Status Design from mixed method research as our research methodology. From quantitative method, survey as the dominant method used. The purpose of survey was to validate the challenges by practitioners, identified through SLR. From qualitative method, we used expert review to evaluate the effectiveness of framework proposed in this research.

CHAPTER 4

ANALYSIS AND RESULTS

4.1 SLR Results

The following section shows the detail of result found by SLR process.

4.1.1 Search Results

After the identification of detailed review protocol including keywords and search strings, we started our search to find the articles relevant to our research topic in five different data sources. We have used advance search option by using our search strings by searching titles, abstracts, and keywords of articles. The objective of our research was to identify the challenges confronted by teams in GSD and distributed scrum and since considerable amount of work is done in GSD and scrum, so we did limit the publication time of the articles and selected the articles published from year 2005 to onward. Pre-defined inclusion/exclusion criteria, and different screenings of found articles performed to extract the most relevant articles for our research. Table 4.1 shows the list of extracted results from different databases.

Table 4.1: List of extracted Results

Sr. No	Database Sources	Initial Screening	1st Filter	2nd Filter	Selected Articles
1	IEEE	61	48	36	23
2	ACM	162	23	11	9
3	Springer	52	39	10	6
4	Wiley's	96	80	39	5
5	Elsevier	97	21	15	5
6	Others	2,880	151	33	8
	Total	3348	362	144	56

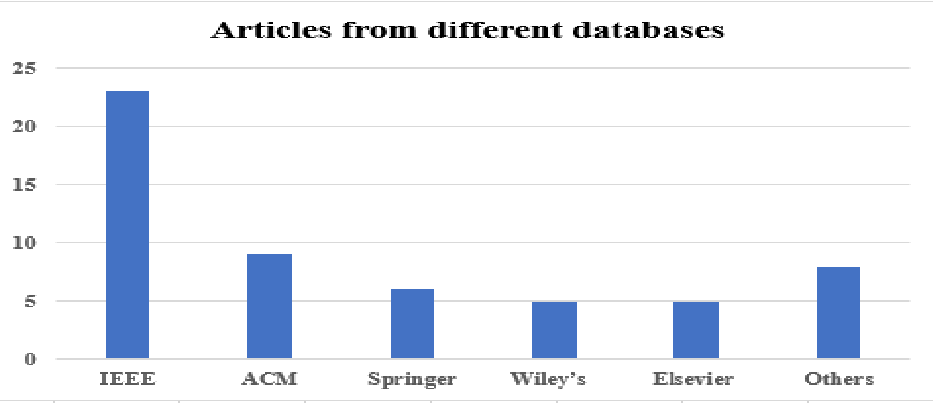

Figure 4.1: Source of Articles

Fig 4.1 shows the source of articles and number of articles found form each source. Others refers to the source of articles obtained while snowball sampling, Google Scholar, or any academic forums etc. Fig 4.2 shows the number of research types with their corresponding percentage used in our research. Fig 4.3 shows the publication years of article found through SLR.

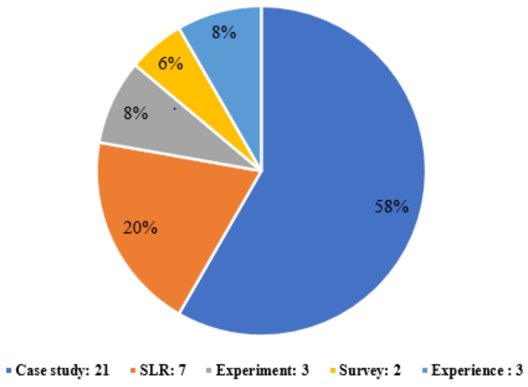

Figure 4.2: Type of Research Articles

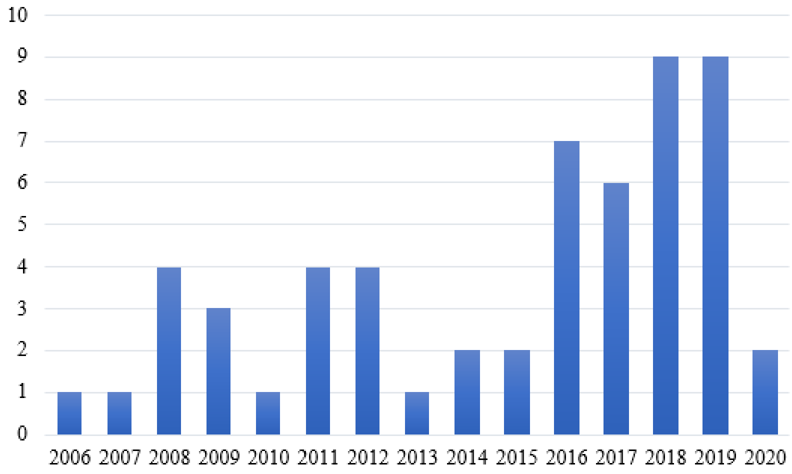

Figure 4.3: Year of publication of Articles.

4.1.2 Quality Assessment Results

After extracting the articles form databases quality assessment performed. We used a checklist based on three questions mentioned in section 3.8.1, to assess the quality of selected articles. We discarded articles which did not fulfil the criteria. Total 56 articles got selected in final assessment.

4.1.3 Data Extraction and Synthesis Results

Before extracting data form selected articles, we discovered the following in initial screenings:

i. Every article has mentioned more than one challenge.
ii. Many articles have described the same kind of challenges in different ways or used different names for same kind of challenges.
iii. Only few articles have classified or categorized the identified challenges and we found those articles with classified data are clearer and easier to understand for us.

We started extracting data by reading the articles selected after quality assessment and identified and extracted different challenges mention in articles along the frequency count of each challenge to find out for how many times each challenge is discussed in articles. We identified total 15 challenges. Table 4.2 shows the challenges identified through SLR.

Table 4.2: List of identified Challenges.

NO	**Challenges**	**References**	**Frequency N= 56**	**Percentage (%)**
1	Communication, Coordination & Collaboration	[5], [8], [9], [2], [10-20], [4], [21], [7], [22-27], [29-34], [36-44], [46-57].	51	91%
2	Knowledge Sharing and Management	[10], [15], [6], [18], [22], [23], [28], [29], [30], [38], [39], [50], [51], [53], [54], [56]	16	28%
3	Project Management	[9], [6], [7], [22], [34], [50], [52], [54], [39], [46], [48], [57], [21], [48]	14	25%

4	Technical Issues	[9], [10], [11], [6], [18], [7], [22], [29], [30], [36], [34], [50], [42], [47], [55]	15	28%
5	Trust Issues	[10], [13], [29], [36], [43], [47], [48], [54], [23], [25], [34], [9]	12	22%
6	Requirement Engineering	[6], [18], [21], [39], [44], [45], [46], [48], [49], [50], [24]	11	19%
7	Quality Assurance	[6], [28], [35], [44], [49], [52], [50]	7	12%
8	Configuration / Integration Management Issues.	[10], [15], [6], [18], [28], [44], [45], [50]	8	14%
9	Resistance to change.	[15], [44], [45], [35]	4	7%
10	Risk Management	[10], [33]	2	3%
11	Architectural Issues	[48], [51], [6], [10], [34], [46], [49]	7	12%
12	Shared understanding	[9], [16], [29], [54], [56]	5	9%

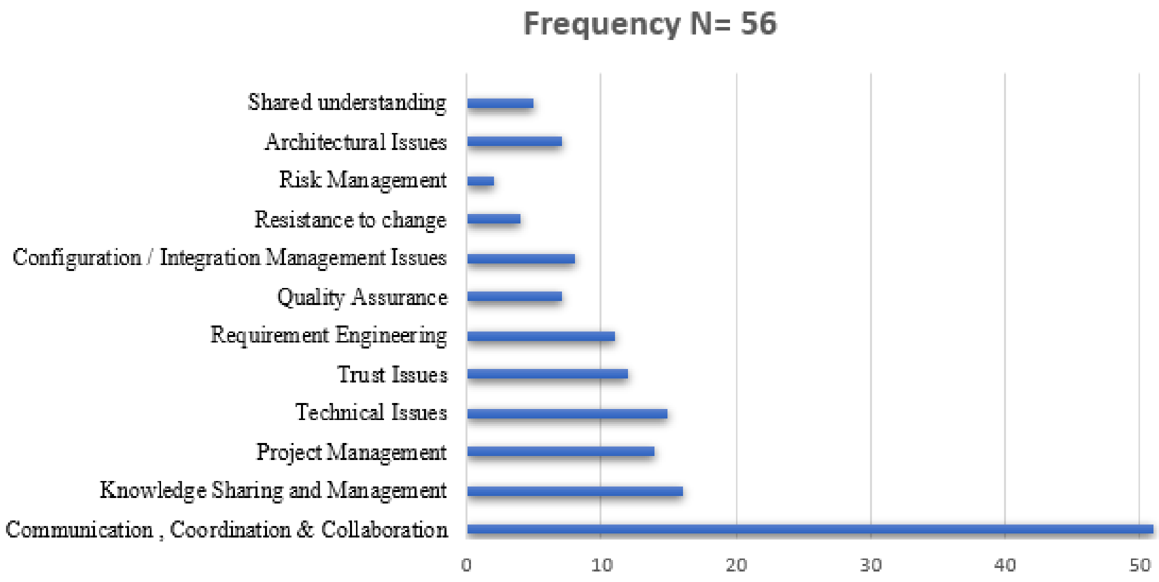

Figure 4.4 Percentage of identified Challenges

Fig 4.4 shows the percentage of each challenge confronted by teams in distributed scrum and from SLR results it is clearly shown that effective communication, co-ordination, and Control among teams is the most challenging factor while using distributed scrum.

4.2 Survey Results

After conducting SLR, we got the list of challenges confronted by distributed scrum teams. To validate the data identified through SLR we conducted an online survey from distributed scrum practitioners using google forms. A questionnaire is designed by using the challenging factors identified through SLR.

Total 305 respondents responded to the survey. To get to know about the background of respondents, we asked few demographic questions and from results it is found that every respondent had an experience in distributed scrum.

We used 5-point likert scale to ask the core question and two types of likert scale are used for that. For first type of likert scale agreement items such as “Strongly Agree”, “Agree”, "Neutral", "Disagree" and "Strongly Disagree" were used to find out the opinion of respondents about certain challenges, covered 11 questions. Another likert scale used in survey contained frequency items such as "Always", "Frequently", "Sometimes", "Seldom" and "Never" in order to find out how many times respondents had confronted some certain challenges and covered 6 questions.

4.2.1 Respondents Profile

To validate the challenge identified through SLR a survey is conducted from distributed scrum practitioners and asked demographic questions to find out about their background. Results shows that all the respondents had experience in distributed scrum. Fig 4.5 shows the years of experience of respondents in distributed scrum.

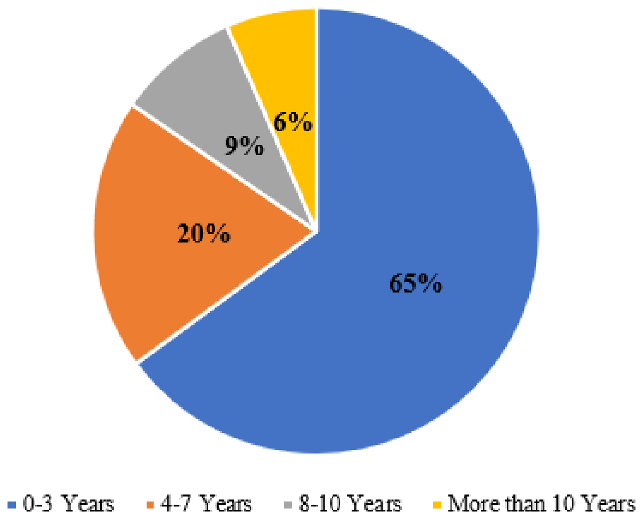

Figure 4.5: Percentage of Respondent's Experience

Table 4.3: Result of Responses from Survey

No	**Factors**	**Strongly Agree (2)**	**Agree (1)**	**Neutral (0)**	**Disagree (-1)**	**Strongly Disagree (-2)**	**Total (305)**
1	Communication, Coordination & Collaboration	114*2= 228	135*1= 135	33*0= 0	20*-1= -20	3*-2= -6	337
2	Technical Issues	105*2= 210	153*1= 153	33*0= 0	12*-1= -12	2*-2= -4	347
3	Configuration / Integration Management	35*2= 70	142*1= 142	85*0= 0	42*-1= -42	1*-2= -2	168
4	Risk Management	55*2= 110	158*1= 158	57*0= 0	31*-1= -31	4*-2= -8	229
5	Quality Assurance	63*2= 126	136*1= 136	46*0= 0	54*-1= -54	6*-2= -12	196
6	Lack of resistance to Change	59*2= 118	149*1= 149	63*0= 0	30*-1= -30	4*-2= -8	228

Table 4.4 Result of Responses from Survey

No	**Factors**	**Always (4)**	**Frequently (3)**	**Sometimes (2)**	**Seldom (1)**	**Never (0)**	**Total (305)**
1	Software Architectural Understanding	24*4= 96	96*3= 288	149*2= 298	31*1= 31	15*0= 0	713
2	Knowledge Sharing and Management	27*4= 108	85*3= 255	132*2= 264	46*1= 46	13*0= 0	673
3	Project Management	36*4= 144	72*3= 216	110*2= 220	67*1= 67	20*0= 0	647
4	Requirement Engineering	31*4= 124	94*3= 282	121*2= 242	47*1= 47	12*0= 0	695
5	Shared Understanding	38*4= 152	82*3= 246	120*2= 240	43*1= 43	22*0= 0	681
6	Trust Issues	42*4= 168	53*3= 159	100*2= 200	52*1= 52	58*0= 0	579

Through survey quantitative data got collected against each challenging factor. Analysis on the collected data has performed in order to accept or reject the factors. Table 4.5 and 4.6 shows the result from responses against each factor.

4.2.2 Results from Weightage Values

Weightage value are the values that talks about the average response collected against each factor. These values are important in order to take a decision about the acceptance or rejection of each factor. Here we have calculated the average weightage value of each factor by using function of Mean.

Avg weightage response = Weightage Value / Total No of responses

Table 4.7 shows the average weightage response against each factor along with the results such as accepted or rejected.

Table 4.5: Accepted or Rejected Results.

No	**Factors**	**Weightage Values**	**Avg. Weightage Responses**	**Results**
1	Communication, Coordination & Collaboration	337	337/305 = 1.13	Accepted
2	Technical Issues	347	347/305 = 1.10	Accepted
3	Configuration / Integration Management	168	168/305 = 0.55	Rejected
4	Risk Management	229	229/305 = 0.75	Rejected
5	Quality Assurance	196	196/305 = 0.64	Rejected
6	Lack of resistance to Change	228	228/305 = 0.74	Rejected
7	Software Architectural Understanding	713	713/305 = 2.33	Accepted
8	Knowledge Sharing and Management	673	673/305 = 2.20	Accepted

9	Project Management	647	647/305 = 2.12	Accepted
10	Requirement Engineering	695	695/305 = 2.27	Accepted
11	Shared Understanding	681	681/305 = 2.23	Accepted
12	Trust Issues	579	579/305 = 1.89	Accepted

4.2.3 Result in sequence

After finding the average weightage response results some factors got accepted and others got rejected based on their average weightage results. The factors with average value greater or equal to 0.90 for 1st type of likert scale got accepted and from 2nd likert scale factors with average value greater or equal to 1.50 got accepted and remaining factors got rejected. Table 4.8 shows the accepted and rejected factors in sequence as final survey results.

Table 4.6: Accepted or Rejected Results in Sequence

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

Out of 12 factors 8 factors including software architectural understanding, requirement engineering, shared understanding, knowledge sharing and management, project management, trust issues, lack of collaboration, communication & coordination and technical issues have values greater or equal to 0.90 and 1.50 according to our defined criteria, so these factors got accepted. Remaining 4 factors including risk management, lack of resistance to change, quality assurance and configuration/ integration management with values less than defined criteria got rejected. Fig 4.6 shows the average weightage response against each factor.

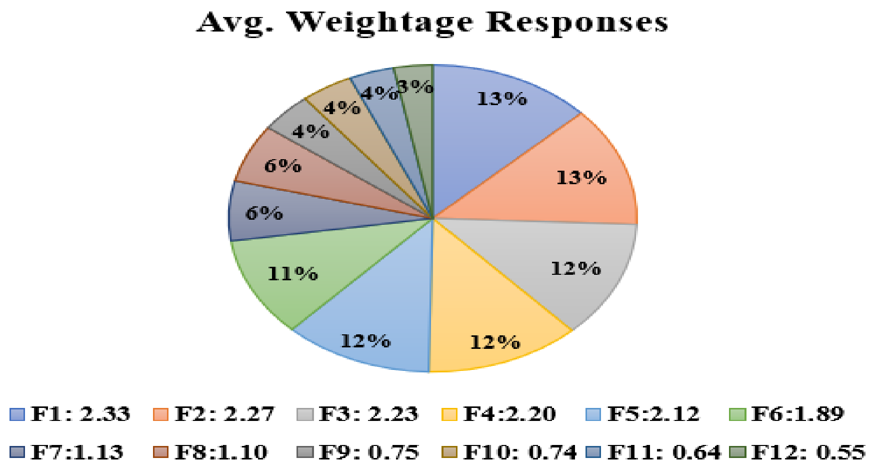

Figure 4.6: Average weightage response against each Challenge

4.3 Results Explanation

While doing analysis on survey results, we set the criteria that only those factors will be accepted that will have value greater than 0.90 in 1st type of likert scale and value greater than 1.5 in 2nd type of likert scale in average weightage response. Based on our defined criteria total 8 factors from F1 to F8 out of 12 factors got accepted as their average response was greater than 0.90 and 1.50 according to our defined criteria and remaining factor from F9 to F12 got rejected due to their lower average response according to defined criteria.

To evaluate the reliability and internal consistency of our results collected through survey, we calculated Cronbach alpha's value in excel. As we have used two types of likert scales in our survey so two different Cronbach alpha's values have calculated. Cronbach alpha's value for first type of Likert scale is "0.85" that covers the responses of 6 factors and for second type of likert scale used in our research Cronbach alpha's values is "0.86" and it covers the responses of remaining 6 factors.

4.3.1 Cronbach Alpha

To evaluate the consistency or reliability of our scales used in research to collect data through survey it is important to calculate the Cronbach alpha's value. It helped us to identify that the scale used to collect data is consistent and we can rely on the results collected through our used scales.

Cronbach alphas value greater than "0.70" is considered as reliable value. As we have used two different types of likert scales so two different Cronbach alphas values have calculated. The first calculated value of Cronbach alpha is "0.85" and it covers 6 factors from the total 12 factors. Remaining 6 factors are covered by another type of likert scale and the calculated value for this scale against 6 factors in "0.86".

4.3.2 Low significance factors

Total 5 factors out of 12 factors got reject due to their lower average weightage value. Rejected factors includes risk management, lack of resistance to change, continuous integration, quality assurance and configuration management.

i. Risk management with average value 0.75 got rejected as most of the respondents have responded to neutral (neither agree nor disagree) option.
ii. Lack of resistance to change with value 0.74 is rejected as it does not fulfil our acceptance criteria and have average value lower than 0.80.
iii. Quality assurance is the 3rd rejected factor with value 0.64. Reason behind the rejection of this factor is that respondents did not agree that quality assurance is a challenge in distributed scrum development.
iv. Configuration/ Integration management with value 0.55 got rejected because most of the respondents marked it as neutral.

4.3.3 High Significance factors

Out of 12 factors asked in survey total 8 factors are accepted. Factors accepted from 1st likert scale includes communication, coordination & collaboration and factors accepted from 2nd likert scale includes software architectural understanding, requirement engineering, shared understanding, knowledge sharing and management, project management, and trust issues.

i. Communication, Coordination & Collaboration is an accepted factor with value 1.13. It is accepted because most of the respondents marked it as strongly agree or agree as they find it is the biggest challenge in distributed scrum.
ii. Technical issue with value 1.10 is another accepted factor. It is accepted because respondents were agreed that teams have to face many technical issues in distributed development
iii. Software Architectural Understanding is the 1st accepted factor form 2nd likert scale. Average value of this factor is 2.33. Respondents were agreed that their teams have frequently faced software architectural understanding issues.

iv. Requirement Engineering with average value 2.27 is another accepted factor from 2nd type of likert scale. Respondents responded that sometimes their team must face requirement engineering issues in distributed development.

v. Shared Understanding with value 2.23 got accepted because respondents reported that sometimes their teams must face shared understanding issues.

vi. Knowledge Sharing and Management with value 2.20 is an accepted factor. Survey results against this factor reported that teams have faced this issue many times during development.

vii. Project Management with average value 2.12 got accepted. Results from respondents revealed that sometimes development teams must face this challenge and it needs to be resolved.

viii. Trust issues is that last accepted factor from 2nd likert scale. Its average value is 1.89. Most of the results against this factor revealed that sometimes teams must face this issue and it needs to be resolved.

4.4 Explanation of Accepted Factors

Factors accepted by practitioners through survey are as follow:

4.4.1 Communication, Coordination & Collaboration

For every type of development despite of its nature either dispersed or co-located communication among teams and stakeholders is very important [40]. It is a way to exchange information with each other either formally or informally [83]. For software development communication is one pf the critical and fundamental process specially for distributed teams[84]. Communication between product owner and stakeholders helps to understand the interests of stakeholders clearly and design the backlog accordingly to convey the interests of the stakeholders to teams. Communication among development team and scrum master helps to track the performance of teams and to ensure that team is moving in right direction by following the scrum practices. Communication among development team members helps to build trust upon each other by sharing progress of work and to understand each other to gain success in every sprint and in whole project. Basically, Scrum highly relies on face-to-face direct communication so, communication among everyone who is a part of project either stakeholders or product owners or scrum master or development teams plays a positive role in the success of project. But the team who are distributed in different locations either locally or globally have bought huge challenges related to communication due to geographical, socio-cultural, or temporal differences. Challenges related to communication lead to many other challenges including lack of coordination and collaboration among teams, lack of trust, shared understanding, software architectural understanding etc. Result of SLR shows that 91% of studies have discussed communication challenges and stated that lack of coordination and collaboration among teams is due to communication issues among teams. As most of the papers have discussed communication, coordination, and collaboration challenges together so we also have merged these two challenges to address their mitigation strategies.

4.4.2 Software Architectural Understanding

Software architectural understanding is another significant challenging factor in distributed scrum identified from survey results with average weightage value 2.33 that needs to be resolved. The main reasons behind misunderstood or unclear software architecture, found from literature is unmanaged architectural changes. Usually, the architecture of software is based on 2 to 3 critical requirements and the sudden changes in requirements, as agile methods welcome new changes at any stage of development, might changes the priority of critical requirements leads to the change of software architecture. To address the changes in requirements to whole development team is difficult task especially in distributed scrum where team is dispersed globally with different working hours. Another reason behind misunderstood software architecture is that developers might understand critical requirements differently due to lack of communication and unsynchronized working hours. To equally understand critical requirements by all teams, it is important to clearly communicate and discuss those critical requirements with development teams.

4.4.3 Requirement Engineering

Requirement engineering plays a vital role in the success of any project. Even though agile methodology and requirement engineering looks incompatible as requirement engineering is about heavy documentation and agile methodology specifically scrum method is people oriented that does not demand heavy document but still scrum can get enough benefit from documentation especially while working in distributive manner [12]. Different articles have reported the issues regarding requirements engineering in terms of distributed scrum development including misunderstood requirements due lack of integration between user stories and designed use cases, due to unclear or ambiguous requirements, requirement traceability issues due to lack of unified backlog, unclear requirement priorities and requirement change management issues [85].

4.4.4 Shared Understanding

Shared understanding has been identified as critical challenge from literature specially is distributed scrum development. It is very important for all the development teams to equally share and understand the things they are working on to make a project success [86]. There are different reasons associated with this challenge to cause this challenge including language barriers, lack communication and collaboration among teams, lack of transparency in shared data or the shared knowledge is ambiguous, lack of trainings about different cultures and languages as teams are working in distributive manners among the teams people with different cultural backgrounds and lack of knowledge and trainings about the tools and technologies that are being used by the teams for development [41][19].

4.4.5 Knowledge Sharing and Management

Sharing or exchange of knowledge among distributed teams is always a challenging task due to lack of direct face-to-face communication among team members. Reasons behind the occurrence of this challenging factor includes tacit knowledge (untold requirements resides in customers mind by considering them obvious) , lack of communication, language barriers, temporal differences, technological issues(lack of unified repository/ backlog, lack of knowledge management tools), misunderstood shared knowledge[87][41]. From literature it is found that effective and sufficient communication, coordination, and collaboration among distributed teams is a key to resolve knowledge sharing and management issues.

4.4.6 Project Management

Project management is a difficult task in software development and to manage distributed project is even more difficult because in distributed project new challenges i.e., temporal distance, cultural differences, language barriers, geographical distance etc. that effects communication and coordination, trust, problem solving and many other factors that impacts on the success of project, are added in already complex project [88]. In literature several challenging factors are reported that are associated with the challenges of project management including ineffective communication and coordination issues due to temporal and cultural differences, lack of trust, different level of knowledge sharing and management, people management, roles and responsibilities identification, synchronization in working hours, and performance visibility [89].

4.4.7 Trust Issues

Trust plays an important role to ascertain the success of any agile or non-agile project. Trust among team members helps them to collaborate with each other in a better way, leads to better performance by teams. As scrum teams are self-organizing and collaborative in nature so the importance of trust among teams became higher for scrum. But the trust among distributed scrum teams is affected due to lack of face-to-face communication and collaboration among teams as scrum is enabled on great communication and collaboration between teams. From literature it is revealed that there are various reasons behind lack of trust among teams including lack of communication and coordination between team members due to cultural differences, poor bonding with team members due to ineffective communication and unsynchronous working hours, no sense ownership and belonging to teams. All these causes leads to lack of trust among team members and have consequences on teams such as lack of commitment for work by team members, poor team performance and conflicts among teams[43].

4.4.8 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 teams. 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][54]. Technical issues can be mitigated by focusing on all other challenges identified in this study.

4.5 Summary

In this chapter we have discussed the results of SLR we have conducted earlier in chapter 2. All the challenging factors identified through SLR are mentioned in a table. Moreover, analysis on survey results is performed in order to accept or reject the factors that are identified through SLR. We have defined or acceptance and rejection criteria on based on those criteria some factors which fulfilled our defined criteria got accepted and remaining factors got rejected. After acceptance and rejection of each factor we have explained the reason of acceptance and rejection of each factor with average weightage value against each factor. After explaining the reason behind acceptance and rejection against each factor we have discussed the accepted factors in detail in order to get deep insight of those factors.

CHAPTER 5

DISCUSSION

5.1 Overview

In our study, we conducted systematic literature review to identify the challenges confronted by distributed scrum teams. Then we conducted an industrial survey from distributed scrum practitioners to validate those challenges identified through SLR. After validation of the challenges, we reviewed the selected articles to find out what are the mitigation or resolution strategies adopted by teams to overcome the confronted challenges. Based on those mitigation or resolution strategies, we developed a framework to mitigate the identified challenges and conducted an expert review to evaluate the effectiveness of our proposed framework.

5.2 Conceptual 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.

5.3 Framework Development Process

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

- To identify the components of our conceptual framework, we reviewed the several frameworks discussed in our selected studies i.e.([7],[9],[11], [22], [23]).
- SLR is conducted to identify the challenges confronted by distributed scrum teams and then survey is conducted from practitioners to validate those identified challenges.
- Review 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.

5.4 Components of Framework

Our framework is broadly composed of 2 main components:

i. Major Challenges
ii. Mitigation Strategies

Figure 5.1 shows the framework we proposed. We discuss each challenge along the mitigation strategies to reduce or overcome these challenges.

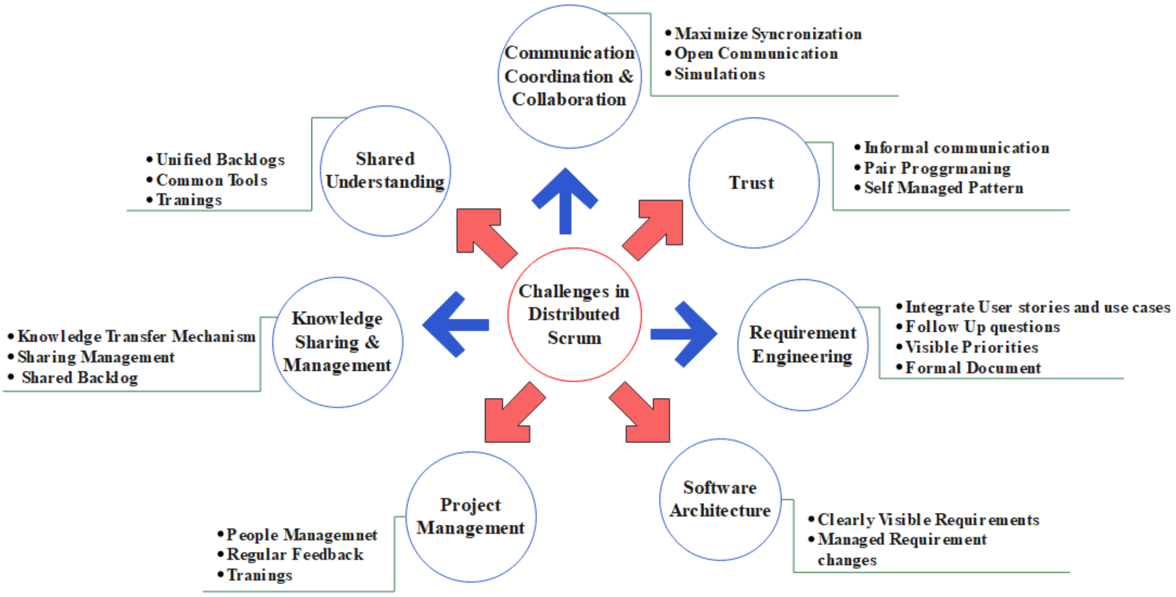

Figure 5.1: Framework for effective utilization of Distributed Scrum

5.4.1 Communication Coordination & Collaboration

As scrum is enabled on great communication, coordination, and collaboration among team, so it is the biggest challenge identified through literature and survey as well. Coordination and collaboration among teams depends upon communication so, communication is the key for better coordination and collaboration among teams [87]. Different resolution strategies are being by different teams to overcome these issues. We have identified some best strategies to overcome these issues.

i. **Maximize Synchronized working hours:** This is widely used strategy in distributed scrum development to improve effective communication among teams. As teams are working from different location, with different working hours 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 [23][7][34][40][90][91][92][93].

ii. **Open Communication:** Open communication is another practice identified from literature to enhance coordination and collaboration among teams. It allows the teams members to openly discuss their views about something and allowed to add suggestions in something if required that leads to the better coordination and collaboration among teams [34][94][93]. This strategy is identified from very few articles with effective results so, it needs to be adopted more often to enhance team performance.

iii. **Simulation:** As scrum require face-to-face communication, which is not possible in distributed environment, so project owners try to replace direct face-to-face communication with rich communication channels through simulations. By providing visualizing environment through simulation with high speed and quality simulation tools for face-to-face discussions helps the teams to communicate effectively leads to better coordination and collaboration among distributed teams [14], [22], [95].

5.4.2 Trust

Trust is another significant challenge reported in literature. Different factors are identified from 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 literature includes:

i. **Informal Communication:** Informal communication is the strategy adopted by different teams build to trust among team members. By adopting different modes of informal communication (i.e., individual or teleconferences, video conferences, emails, instant messages) with formal documentations can help to build trust among team members [86][91][21][37].

ii. **Pair Programming:** It is another strategy adopted by teams in order to build trust among distributed teams is, to encourage pair programming among team. When team members collaborate with each other to share things and experiences, it will positively affect the overall performance of teams [93][96].

iii. **Self-managed patterns:** To build trust among team allow them to work according to their own working pattern and give them the responsibilities. It leads to develop a sense of ownership among teams, results in better performance of teams[97][38][34].

5.4.3 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 it is reported from literature that the team who are working in distributed scrum are facing different challenges associated with requirement engineering due to many reason including lack of formal documentation, invisible priorities of requirements, ambiguous and unclear requirements etc. The best resolution strategies we have identified from literature to resolve the requirement engineering issues includes:

i. **Integrate user stories and use cases:** Form literature it is found that it is very difficult and time consuming, for the development teams to extract user stories and use cases from different tools and then synchronize both to access the record about the completion of work. An unsynchronized user stories and use cases leads to misunderstanding among development teams about work progress and completion of work and especially when teams are working in distributed environment. To avoid misunderstanding about work progress it is important to integrate user stories with use case before starting actual development [24][21][93].

ii. **Follow-up questions:** As teams are working in distributed manner with different cultures and geographic location so the chances of requirements misunderstanding are more, as the person or team member who is sitting far away might have a completely different picture of functionality of some requirements than a product owner. Form literature it is found that some teams have faced the issues of unnoticed misunderstood requirement until the sprint review. To overcome this issue, it is important to ask follow-up question to verify the correct understanding of requirements. Few studies have practiced this follow up questions strategy and results revealed the improvements in requirement understanding process [10] [51].

iii. **Visible Requirement Priorities:** It is identified from literature that different distributed teams are facing the issues in requirement prioritization. Changing priority of requirements leads to changing in software architecture results in software architectural understanding issues. To resolve these issues practitioners suggested to finalize the requirement and prioritize them clearly before starting the actual development [34] [6].

iv. **Formal Document:** Although agile methodologies do not require heavy documentation, but it is found from literature that lack of documentation in distributed scrum development lead 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 standard template understandable to all distributed teams [44][91][98][51].

5.4.4 Software Architecture

Clear understanding of software architecture is very important for the success of any project. Usually, the architecture of any software is based on 3-4 critical requirements [52]. As agile methods welcome changes at any stage of development so teams working in distributed environment find it challenging because sometimes product owner unable to deliver the clearly understandable requirements to the teams due to continuous changes in requirements and with the continuous change in requirements, change in requirements remains unmanaged that leads to software architectural understanding issues. To mitigate this challenge different practices are adopted by different teams. The best practice we identified for our framework includes:

i. **Understanding of architecture drivers:** As architecture of software is based on 3-4 critical requirements and remaining requirements are aligned with those critical requirements. Those critical requirements might be understood differently especially when working in distributed environment. In order to equally understand those critical requirements by all teams, devote multiple iterations 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 and software architectural understanding among teams [12][52][54].

ii. **Managed Requirements changes:** To manage the requirement changes product owners tries to maintain a unified repository with the traceability process for requirement changes, accessible to all development teams, so development 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 [12][99][98][100].

5.4.5 Project Management

Project management itself a complex task and to manage project is distributed environment is even more difficult task due geographical, socio-cultural distance and temporal differences. Different strategies are adopted by product owners to mitigate the project management issues. Some best strategies include:

i. **People management:** It is reported in different articles that to manage project effectively in distributed scrum, it is important to manage people related activities effectively. 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 specific task, and use any tool i.e., to track the progress of the team and the work[89][28][2][88][42].

ii. **Regular feedback:** Get regular feedback from stakeholders about work by regularly communicating with them ensue that the team is progressing in right direction [38][36] [96].

iii. **Trainings:** By providing trainings or coaching about work whenever is demanded or required by teams to learns the common skills by all distributed teams helps the teams to timely deliver the exact things demanded by customers. By allowing or sponsoring the teams to attend the training organized by another organization helped the teams to enhance their skills in latest technology results in, better performance from teams [48][29].

5.4.6 Knowledge Sharing and Management

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

i. **Sharing management:** The issues confronted by teams due to many reasons including lack of unified backlog and inundated backlog. 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 accessible to all the teams can helps the teams to resolve this knowledge sharing and management issues as common data will be visible to all teams [38][11][87] [53][18][41].

ii. **Standard mechanism:** By defining a proper mechanism and standard template to share data (i.e. standard template for project documentation for all the teams) with each other among distributed teams can help to over the knowledge sharing and management issues[24][31][18][101].

5.4.7 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 leads to project failure. Some best practices identified from literature to ensure shared understanding includes:

i. **Trainings:** As teams are working in globally distributed environments and belongs to different cultures so, language barrier is the big reason behind lack of shared understanding as team members who do not understand similar language will hesitate to communicate and collaborate with other members in teams. So, by providing trainings about different cultures and languages can help to resolve this issue [46][7][48].

ii. **Common Tools:** Another reason behind lack of shared understanding among teams is the usage of different tools to work on for similar project, as it is difficult for the distributed teams to understand different tools for similar work. By ensuring the usage of common tools for development by all the teams who are working in similar project can helps the teams to equally understand the data or information shared with them [34][21].

iii. **Unified Backlog:** By maintaining a unified backlog having transparent requirements with clearly visible requirements for all the teams can helps 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 form of completed sprints and current sprints, it helps the teams to understand the work more clearly [19] [21][27] [59].

5.5 Expert Review

We conducted expert review to validate the effectiveness of our proposed framework and interviewed 5 distributed scrum experts. Table 5.1 shows the comments of experts against strategy, proposed in our framework.

Table 5.1 Expert Review

Challenges	Suggested Strategies	P1	P2	P3	P4	P5	Result Summary
Communication Coordination & Collaboration	• Maximize synchronization. • Open communication • Simulations	• By maximizing synchronized working hours of distributed teams, will help to improve communication and coordination among teams. • Encouraging teams to Participate and openly communicate with other teams will helps to improve coordination and collaboration among teams. • Effective simulation tools will help to better coordination among distributed teams	• Maximum synchronization will be effective to mitigate communication and coordination issues • Usage of Good quality simulation tools with permission to communication and discuss about things openly will be very helpful to better communication and coordination among distributed teams	• Synchronized working hours is effectively working for our distributed teams. • Open communication will be helpful to improve coordination and communication among teams • Simulation would be effective to improve coordination among teams.	• Synchronization in working hours of distributed teams is very effective to overcome communication and coordination issues • Encourage distributed teams to communicate openly will help to improve team's coordination and collaboration • Coordination and collaboration among teams could be enhanced by using high quality simulation tools to distributed teams	• Working in synchronized working hours will help to mitigate communication and coordination issues. • Open communication will help to reduce communication gap results in better coordination and collaboration among teams • High quality simulation will be effective to overcome coordination issues.	• Strongly Recommended • Recommended • Recommended
Trust	• Informal communication • Pair programming	• Informal communication among team members will helps teams to understand each other in a better way and will	• Teams will be able to understand each other in a better way and it will be easier for teams to build trust upon each other this way	• To build trust among distributed teams, encouragement towards informal communication	• Informal communication will effectively contribute to build trust upon each other is distributed development	• Informal communication between teams is an effective way to build trust on each other. • Relying on each other for the completion of	• Strongly Recommended • Recommended

		be helpful to build trust on each other. • Understanding of the working pattern of each other by working in pairs will be effective to build trust	• Teams working in pairs would have less trust issues.	will be very effective. • Pair programming would help to mitigate trust issues by depending upon each other for the completion of tasks.	• Trust among teams will be enhances when working together and depending upon each other to complete a task.	any task and taking suggestion from each other when required will help teams to build trust on each other.	
Requirement Engineering	• Integrate User stories and Use cases. • Follow-Up Questions • Visible Priorities • Formal Document	• Integration between US and UC will help to clearly understand needs and demands. • Asking follow-up question from stakeholders will be effective to clearly understand the requirements. • Finalizing requirement priorities before starting actual development will effectively contribute to deliver exactly required product. • Formal documentation will help the teams to avoid any	• Getting US and UC in common tools will help the teams to understand requirements more effectively • It will help ensure that the requirements are clearly understood by development teams • Teams will be clearer about what they need to deliver in each sprint by setting visible priorities at start • To avoid any misunderstanding at any stage of	• Integration between US and UC would be very effective, to clarify the requirements going to develop. • Follow up questions are important to assure that requirements are clear to development teams • Clearly defined and visible priorities of requirements is very important to deliver desired project • Formal document will be very helpful	• To get a clear picture of what is going to develop it is important to integrate user stories with user stories • Follow up questions are important to assure that requirements are understood clearly to the development teams • Clearly visible priorities help the teams to deliver exactly required project on time • Formal document is important to avoid	• Integration between US and UC will help teams to avoid any requirement misunderstanding while development • To ensure that requirements are clearly described and understood, it is important to ask to follow up questions. • Visible priorities are important to plan and deliver a successful project • Maintain all the work needs to be done in form of formal requirement	• Recommended • Strongly Recommended • Recommended • Strongly Recommended

		misunderstanding about requirements	development formal document is very important.	to avoid any misunderstanding of requirements	any requirement misunderstandings.	document is important to avoid any misunderstanding and problem in later development stages.	
Software Architectural	• Understanding of Architecture Drivers. • Managed Requirement Changes	• clear and understandable core requirements will help the teams to understand software architecture effectively. • Properly managed requirement changes will help to avoid architectural misunderstandings	• Clearer knowledge about core requirement on which software architecture of is based is very important to avoid any loss deliver successful project • Misunderstanding of software architecture can be avoided by properly managing the changing in requirements and requirement change requests.	• To design software architecture, it is very important to clearly understand the architecture drivers on which software architecture is based • To avoid any misunderstanding in software architecture while development, it is important to effectively manage every change in requirement	• Understanding of core requirements on which software architecture is based will help to design effective software architecture • To avoid any architectural problems and misunderstanding, it is important to properly manage the requirements changes	• Understanding of architectural drivers on which the whole software architecture is based, is an effective way to overcome architectural issues • Any changes in requirement should be managed properly to avoid any misunderstanding in development	• Recommended • Recommended
Project Management	• People Management • Regular Feedback • Trainings	• Considering required competencies while assigning responsibilities will helps the scrum master to effectively manage the distributed projects.	• Assigning roles and responsibilities to the only people who fulfil the required criteria, would be easier to manage project	• To effectively manage projects, it is important to consider right competencies while assigning	• Project management will be easier by focusing on people management • Regular feedback will ensure that team	• For effective project management it is important to consider people management while assigning tasks. • Regular feedback from all the	• Strongly Recommended • Strongly Recommended • Recommended

		• Regular feedback from stakeholders is very important to ensure that team is progressing in right direction. • Providing trainings to teams to learn new skills is very important for the competent development.	• Regular feedback from the stakeholders either the development team or the customer is very important to successfully manage the projects. • Teaching new skills to teams and allowing them to learn and grow is very important for competent development.	tasks to team members • Regular feedback from project stakeholders will play a vital role in effective project management • Organizing trainings for distribute teams whenever is demanded or required will help to mitigate project management issues	is working in right direction • Trainings are important to teach new skills to teams for effective development.	stakeholders will help to effectively manage the projects. • Provide trainings and teaching teams about innovative technologies will help to deliver successful projects with right competencies.	
Knowledge Sharing & Management	• Sharing Management • Standard Mechanism	• Unified and regularly updated backlog or repository is very important to share and manage all important knowledge with distributed teams. • Sharing knowledge by predefined standard mechanism will helps the teams to manage the knowledge of data shared with them effectively.	• To deliver a successful project it is very important to properly manage all the data, being shared with all the teams. • Sharing data with all the teams by using same or standard pattern will be very effective for the teams to easily	• Maintaining a proper management system for the knowledge shared among all distributed team will effectively mitigate knowledge sharing and management issues • By adopting a standard	• To effectively manage the data shared with all distributed teams it is important to maintain a knowledge sharing and management system • To mitigate knowledge sharing and management issues it is important to ensure that data is	• It is important to maintain a management system to maintain all the knowledge shared with all the teams to avoid any problems and to keep the record of the completed work and the work need to be done. • Standard mechanism followed by all the teams to shared	• Recommended • Recommended

			understand the shared data.	mechanism for sharing knowledge among all the team will help to resolve knowledge sharing and management issues.	shared among all teams by following standard mechanism**.**	knowledge with each other is important to avoid any problem.	
Shared Understanding	• Unified Backlogs • Common Tools • Trainings	• Unified backlogs accessible to all distributed teams will be very helpful for the teams to get shared understanding of data. • Usage of similar tools by all distributed teams will be very helpful to equally understand the things teams are working on**.** • Providing trainings to teams in about different cultures and languages will helps the teams to mitigate shared understanding issues.	• Maintaining a unified backlog will be very effective to ensure that the same knowledge is being shared with all the distributed teams. • To avoid any development misunderstanding and complexities it will be effective to ensure that all the teams are working on similar tools. • Trainings about different cultures in different language will be effective to resolve shared understanding issues.	• Unified backlog accessible to all teams will assure that similar knowledge is being shared among all distributed teams • Usage of common tools for development will help to avoid any development and integration complexities. • To ensure that shared knowledge is equally understandable by all distributed teams, trainings would be very effective	• Unified backlog will help to ensure that same knowledge is shared and accessible by all teams • To avoid any complexities, it is important to assure that all developments teams are doing development using common tools • Trainings about different cultures and working patterns will help the teams to get equal understanding of the shared knowledge	• Shared and unified backlog accessible to all is important to assure that same data is shared among all distributed teams • Bound teams to work on common tools among all distributed locations will help to avoid many issues and complexities including integration and configuration management. • By providing trainings about different cultures and working patterns will help to mitigate shared understanding issues.	• Strongly Recommended • Strongly Recommended • Recommended

5.6 Summary

In this chapter we have discussed the framework proposed in this study to mitigate the challenges identified through SLR. All the strategies proposed against each challenge discussed in detail that how each strategy will help to mitigate the identified challenges. After detailed explanation of each component of framework, we conducted expert review conducted to evaluate the effectiveness of proposed framework. We have discussed the results from expert review, along the comments each participant gave against every strategy proposed in our framework.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Overview

In this study, we identified the most common challenges encountered by distributed scrum teams by conducting systematic literature review. We conducted an industrial survey to validate the challenges identified through SLR. Next, we proposed a framework to mitigate the validated challenges by consolidating best practices adopted by different teams to overcome the challenges and conducted an expert review to evaluate the effectiveness of proposed framework.

6.2 Reviewing Research Questions

RQ1. What are the challenges confronted by teams during the execution of scrum in distributive manner?

Our first research question was about the identification of challenges confronted by distributed scrum teams. To investigate this research question we conducted in-detailed systematic literature review by following Kitchenham guidelines [76]. We have followed a complete protocol shown in figure to conduct SLR. We conducted snowball sampling to ensure that no relevant article is missed. By strictly following inclusion/exclusion and quality assessment criteria total 56 studies got selected to conduct SLR. We have read selected articles in detail to identify the challenges confronted by distributed scrum team. Key findings of selected studies were extracted and placed into a tabular from for better understanding shown in table 1. From extensive SLR we identified 12 major challenges i.e., Communication, Coordination & Collaboration, Knowledge Sharing and Management, Project Management, Technical Issues, Trust Issues, Requirement Engineering, Quality Assurance, Configuration / Integration Management Issues, Resistance to change, Risk Management, Architectural Issues and Shared understanding Issues along the frequency against each challenge, that how many articles have discussed same challenges shown in table 4.2.

After identification of challenges, we conducted an online industrial survey to validate the challenges identified through SLR by distributed scrum practitioners. We have used two types of likert scales in our survey to collect responses. To evaluate the reliability and accuracy of our survey results, we calculated Cronbach alpha's value. Later, we defined a criterion by evaluating the average weightage response against each challenge, to accept and reject the challenges. Out of 12 challenges identified through survey, 8 challenges got accepted by practitioners i.e., Communication, Coordination & Collaboration, Knowledge Sharing and Management, Project Management, Technical Issues, Trust Issues, Requirement Engineering, Architectural Issues and Shared understanding issues and remaining 4 challenges i.e. Quality Assurance, Configuration / Integration Management Issues, Resistance to change, Risk Management got rejected, shown in table 4.8 along with average weightage response against each challenge.

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

Our second research question was about the identification of mitigation strategies adopted or suggested by distributed scrum teams to overcome the confronted challenges. To answer this research question, we have conducted systematic literature review. Through SLR we identified different mitigation strategies adopted by distributed scrum teams to treat the identified challenges. After that, we consolidated the best practices into a framework from multiple strategies suggested and adopted by distributed teams to mitigate the challenges, validated by practitioners. To evaluate the effectiveness of our proposed framework and to ensure that our proposed framework will help the practitioners to effectively mitigate the confronted challenges, we conducted expert review by following Bilal Ayyub guidelines [82] and conducted interviews of 5 experts working in distributed scrum. Results of expert review found supportive and shows that our proposed framework will efficiently contribute to effective distributed scrum development.

6.3 Research Contribution

The contribution of our study is as follows:

i. Identified the Challenges confronted by distributed scrum teams by conducting an extensive literature review and validated the identified challenges through industrial survey.
ii. Proposed a framework by consolidating the best practices, identified through SLR to effectively mitigate the identified challenges.
iii. Conducted expert review to evaluate the effectiveness of proposed framework.
iv. Proposed framework will help the framework to effectively mitigate the confronted challenges and deliver sustainable projects.

6.4 Limitations

Although we conducted this study by following well defined and thorough research instruments [76][74], following limitations might exist to this study:

i. 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.

ii. We have not conducted any industry-based case study to evaluate our proposed framework.

6.5 Future Work

An industrial case study can be conducted by using the proposed framework to evaluate the further effectiveness of the framework and ensure sustainable distributed scrum development.

REFERENCES

[1] N. H. Arshad and F. A. Hanifah, "Issues and challenges in NSDI implementation," *Int. Conf. Syst. Sci. Simul. Eng. - Proc.*, vol. 3, no. 8, pp. 65–70, 2010.

[2] A. Khalid, S. A. Butt, T. Jamal, and S. Gochhait, "Agile Scrum Issues at Large-Scale Distributed Projects: Scrum Project Development at Large," *Int. J. Softw. Innov.*, vol. 8, no. 2, pp. 85–94, 2020.

[3] A. Srivastava, S. Bhardwaj, and S. Saraswat, "SCRUM model for agile methodology," *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2017*, vol. 2017-Janua, pp. 864–869, 2017.

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

[5] 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.

[6] O. Uludag, M. Kleehaus, C. Caprano, and F. Matthes, "Identifying and structuring challenges in large-scale agile development based on a structured literature review," *Proc. - 2018 IEEE 22nd Int. Enterp. Distrib. Object Comput. Conf. EDOC 2018*, pp. 191–197, 2018.

[7] 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," *Proc. - Asia-Pacific Softw. Eng. Conf. APSEC,* pp. 457–464, 2009.

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

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

[10] I. Zada, S. Shahzad, and S. Nazir, "Issues and Implications of Scrum on Global Software Development," *Bahria Univ. J. Inf. Commun. Technol.*, vol. 8, no. 1, pp. 81–87, 2017.

[11] K. Schwaber and J. Sutherland, "The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework," *SCRUM Inc.*, p. 224, 2011.

[12] N. Ramadan and S. Megahed, "Requirements Engineering in Scrum Framework," *Int. J. Comput. Appl.*, vol. 149, no. 8, pp. 24–29, 2016.

[13] 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 Trans. Antennas Propag.*, vol. 39, no. 3, pp. 309–317, 1991.

[14] Y. Khmelevsky, X. Li, and S. Madnick, "Software development using agile and scrum in distributed teams," *11th Annu. IEEE Int. Syst. Conf. SysCon 2017 - Proc.*, 2017.

[15] H. Amar, P. M. Rafi-ul-Shan, and A. Adegbile, "Towards a 5c theory of communication for scrum-based distributed projects," *Bam2019,* p. 25, 2019.

[16] P. Walimbe, "To Overcome Communication Challenges in Distributed/Virtual Scrum Teams," 2016.

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

[18] F. Almeida, E. Miranda, and J. Falcão, “Challenges and facilitators practices for knowledge management in large-scale scrum teams,” *J. Inf. Technol. Case Appl. Res.*, vol. 21, no. 2, pp. 90–102, 2019.

[19] M. Hummel, C. Rosenkranz, and R. Holten, “The role of shared understanding in distributed scrum development: An empirical analysis,” *24th Eur. Conf. Inf. Syst. ECIS 2016*, 2016.

[20] M. Paasivaara and C. Lassenius, “Scaling scrum in a large globally distributed organization: A case study,” *Proc. - 11th IEEE Int. Conf. Glob. Softw. Eng. ICGSE 2016*, pp. 74–83, 2016.

[21] P. Lous, M. Kuhrmann, and P. Tell, “Is scrum fit for global software engineering?,” *Proc. - 2017 IEEE 12th Int. Conf. Glob. Softw. Eng. ICGSE 2017*, pp. 1–10, 2017.

[22] R. Qureshi, M. Basheri, and A. A Alzahrani, “Novel Framework to Improve Communication and Coordination among Distributed Agile Teams,” *Int. J. Inf. Eng. Electron. Bus.*, vol. 10, no. 4, pp. 16–24, 2018.

[23] E. Hossain, P. L. Bannerman, and D. R. Jeffery, “Scrum practices in global software development: A research framework,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6759 LNCS, pp. 88–102, 2011.

[24] M. Cristal, D. Wildt, and R. Prikladnicki, “Usage of SCRUM practices within a global company,” *Proc. - 2008 3rd IEEE Int. Conf. Glob. Softw. Eng. ICGSE 2008*, pp. 222–226, 2008.

[25] A. Välimäki and J. Kääriäinen, “Patterns for Distributed Scrum — A Case Study,” *Enterp. Interoperability III*, pp. 85–97, 2008.

[26] J. Cho, “Distributed scrum for large-scale and mission-critical projects,” *Assoc. Inf. Syst. - 13th Am. Conf. Inf. Syst. AMCIS 2007 Reach. New Height.*, vol. 1, pp. 399–406, 2007.

[27] A. M. AlMutairi and M. R. J. Qureshi, “The Proposal of Scaling the Roles in Scrum of Scrums for Distributed Large Projects,” *Int. J. Inf. Technol. Comput. Sci.*, vol. 7, no. 8, pp. 68–74, 2015.

[28] S. Lee and H. S. Yong, “Distributed agile: Project management in a global environment,” *Empir. Softw. Eng.*, vol. 15, no. 2, pp. 204–217, 2010.

[29] M. Paasivaara, S. Durasiewicz, and C. Lassenius, “Using scrum in distributed agile development: a multiple case study,” *Proc. - 2009 4th IEEE Int. Conf. Glob. Softw. Eng. ICGSE 2009*, pp. 195–204, 2009.

[30] M. Paasivaara and C. Lassenius, “Scaling scrum in a large distributed project,” *Int. Symp. Empir. Softw. Eng. Meas.*, pp. 363–367, 2011.

[31] N. Sekitoleko, F. Evbota, E. Knauss, A. Sandberg, M. Chaudron, and H. H. Olsson, “Technical dependency challenges in large-scale agile software development,” *Lect. Notes Bus. Inf. Process.*, vol. 179, pp. 46–61, 2014.

[32] R. Sriram and S. K. Mathew, “Global software development using agile methodologies: A review of literature,” *2012 IEEE 6th Int. Conf. Manag. Innov. Technol. ICMIT 2012*, pp. 389–393, 2012.

[33] J. Sutherland, G. Schoonheim, E. Rustenburg, and M. Rijk, “Fully distributed scrum: The secret sauce for hyperproductive offshored development teams,” *Proc. - Agil. 2008 Conf.*, pp. 339–344, 2008.

[34] 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,” *ACM Int. Conf. Proceeding Ser.*, pp. 147–156, 2017.

[35] M. Paasivaara, C. Lassenius, and V. T. Heikkilä, “Inter-team coordination in large-scale globally distributed scrum: Do scrum-of-scrums really work?,” *Int. Symp. Empir. Softw. Eng. Meas.*, pp. 235–238, 2012.

[36] M. Hammad and I. Inayat, “Integrating risk management in scrum framework,” *Proc. - 2018 Int. Conf. Front. Inf. Technol. FIT 2018*, pp. 158–163, 2019.

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

[38] M. Kalenda, P. Hyna, and B. Rossi, “Scaling agile in large organizations: Practices, challenges, and success factors,” *J. Softw. Evol. Process*, vol. 30, no.

10, pp. 1–24, 2018.

[39] E. Therrien, “Overcoming the Challenges of Building a Distributed Agile Organization,” pp. 368–372, 2008.

[40] M. D. Kahya and C. Seneler, “Geographical Distance Challenges in Distributed Agile Software Development: Case Study of a Global Company,” *UBMK 2018 - 3rd Int. Conf. Comput. Sci. Eng.*, vol. 3, pp. 78–83, 2018.

[41] R. K. Gupta and P. Manikreddy, “Challenges in Adapting Scrum in Legacy Global Configurator Project,” *Proc. - 2015 IEEE 10th Int. Conf. Glob. Softw. Eng. ICGSE 2015*, pp. 46–50, 2015.

[42] 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,” *2018 4th Int. Conf. Comput. Commun. Autom. ICCCA 2018*, pp. 1–7, 2018.

[43] S. Dorairaj, J. Noble, and P. Malik, “Understanding lack of trust in distributed agile teams: A grounded theory study,” *IET Semin. Dig.*, vol. 2012, no. 1, pp. 81–90, 2012.

[44] D. M. Szabo and J. P. Steghofer, “Coping Strategies for Temporal, Geographical and Sociocultural Distances in Agile GSD: A Case Study,” *Proc. - 2019 IEEE/ACM 41st Int. Conf. Softw. Eng. Softw. Eng. Pract. ICSE-SEIP 2019*, pp. 161–170, 2019.

[45] S. Esquivel, “Communication Issues in Agile Software Development,” pp. 475–484, 2016.

[46] M. Kajko-Mattsson, G. Azizyan, and M. K. Magarian, “Classes of distributed Agile development problems,” *Proc. - 2010 Agil. Conf. Agil. 2010*, pp. 51–58, 2010.

[47] K. Dikert, M. Paasivaara, and C. Lassenius, “Challenges and success factors for large-scale agile transformations: A systematic literature review,” *J. Syst. Softw.*, vol. 119, pp. 87–108, 2016.

[48] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen, “Large-scale agile transformation at Ericsson: a case study,” *Empir. Softw. Eng.*, vol. 23, no. 5, pp.

2550–2596, 2018.

[49] 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,” *ACM Int. Conf. Proceeding Ser.*, pp. 158–163, 2019.

[50] 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, no. December 2017, pp. 161–180, 2018.

[51] S. S. Hossain, *Challenges and Mitigation Strategies in Reusing Requirements in Large-Scale Distributed Agile Software Development: A Survey Result*, vol. 998. Springer International Publishing, 2019.

[52] W. Alsaqaf, M. Daneva, and R. Wieringa, “Quality requirements challenges in the context of large-scale distributed agile: An empirical study,” *Inf. Softw. Technol.*, vol. 110, no. July 2018, pp. 39–55, 2019.

[53] 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,” *J. Softw. Evol. Process*, vol. 30, no. 11, pp. 1–19, 2018.

[54] 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,” *Empir. Softw. Eng.*, vol. 23, no. 1, pp. 490–520, 2018.

[55] A. M. Razavi and R. Ahmad, “Agile development in large and distributed environments: A systematic literature review on organizational, managerial and cultural aspects,” *2014 8th Malaysian Softw. Eng. Conf. MySEC 2014*, pp. 216–221, 2014.

[56] 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,” *Proc. - 11th IEEE Int. Conf. Glob. Softw. Eng. ICGSE 2016*, pp. 84–88, 2016.

[57] 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," *Proc. 2017 Int. Conf. Commun. Comput. Digit. Syst. C-CODE 2017*, pp. 228–232, 2017.

[58] E. Hossain, P. L. Bannerman, and R. Jeffery, "Towards an understanding of tailoring scrum in global software development: A multi-case study," *Proc. - Int. Conf. Softw. Eng.*, pp. 110–119, 2011.

[59] H. Holmström, B. Fitzgerald, P. J. Ågerfalk, and E. Ó. Conchúir, "Usage of SCRUM practices within a global company," *Inf. Syst. Manag.*, vol. 23, no. 3, pp. 7–18, 2006.

[60] S. Shafiq, Y. Hafeez, S. Ali, N. Iqbal, and M. Jamal, "Towards Scrum Based Agile Framework for Global Software Development Teams," *Mehran Univ. Res. J. Eng. Technol.*, vol. 38, no. 4, pp. 979–998, 2019.

[61] M. Paasivaara, C. Lassenius, D. Damian, P. Raty, and A. Schroter, "Teaching students global software engineering skills using distributed Scrum," *Proc. - Int. Conf. Softw. Eng.*, pp. 1128–1137, 2013.

[62] J. Carver, C. Seaman, and R. Jeffery, "Using qualitative methods in software engineering," *... Empir. Softw. Eng. ...*, 2004.

[63] S. Kamolson, "Fundamentals of quantitative research Suphat Sukamolson, Ph.D. Language Institute Chulalongkorn University," *Lang. Inst.*, p. 20, 2007.

[64] M. Kasunic, "Designing an effective survey. Handbook. Carnegie Mellon University," *Softw. Eng. Inst.*, no. September, p. 140, 2005.

[65] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting Empirical Methods for Software Engineering Research Guide to Advanced Empirical Software Engineering," *Guid. to Adv. Empir. Softw. Eng.*, pp. 285–311, 2008.

[66] D. I. K. Sjøberg, T. Dybå, and M. Jørgensen, "The Future of Empirical Methods in Software Engineering Research," *FoSE 2007 Futur. Softw. Eng.*, pp. 358–378, 2007.

[67] S. A. K. Gahyyur, S. Arif, and Q. Khan, "Requirements Engineering Processes, Tools/Technologies, & Methodologies," *Int. J. Rev. Comput. (IJRIC), ISSN 2076-3328*, vol. 2, pp. 41–56, 2010.

[68] D. Spencer and T. Warfel, “Card sorting: a definitive guide,” *Boxes and arrows*. p. 2, 2004.

[69] C. L. Paul, “Investigation of Applying the Delphi Method to a New Card Sorting Technique,” vol. 2006, no. rev 1, pp. 1–19, 2007.

[70] H. Taherdoost, “Sampling Methods in Research Methodology; How to Choose a Sampling Technique for Research,” *SSRN Electron. J.*, vol. 5, no. 2, pp. 18–27, 2018.

[71] S. Jamshed, “Qualitative research method-interviewing and observation,” *J. Basic Clin. Pharm.*, vol. 5, no. 4, p. 87, 2014.

[72] S. N. Khan, “Qualitative research method - Phenomenology,” *Asian Soc. Sci.*, vol. 10, no. 21, pp. 298–310, 2014.

[73] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empir. Softw. Eng.*, vol. 14, no. 2, pp. 131–164, 2009.

[74] American Journal of Sociology, “Bridging the Qualitative-Quantitative Divide: Guidelines for Conducting Mixed Methods Research in Information Sys,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.

[75] N. L. Leech and A. J. Onwuegbuzie, “A typology of mixed methods research designs,” *Qual. Quant.*, vol. 43, no. 2, pp. 265–275, 2009.

[76] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in SE,” *Guidel. Perform. Syst. Lit. Rev. SE*, pp. 1–44, 2007.

[77] T. Meline, “Selecting Studies for Systemic Review: Inclusion and Exclusion Criteria,” *Contemp. Issues Commun. Sci. Disord.*, vol. 33, no. Spring, pp. 21–27, 2006.

[78] T. Dybå and T. Dingsøyr, “Strength of Evidence in Systematic Reviews in Software Engineering,” *ESEM’08 Proc. 2008 ACM-IEEE Int. Symp. Empir. Softw. Eng. Meas.*, no. 7465, pp. 178–187, 2008.

[79] Y. Zhou, H. Zhang, X. Huang, S. Yang, M. A. Babar, and H. Tang, “Quality assessment of systematic reviews in software engineering: A tertiary study,”

ACM Int. Conf. Proceeding Ser., vol. 27-29-Apri, 2015.

[80] C. Wohlin, “Guidelines for snowballing in systematic literature studies and a replication in software engineering,” *ACM Int. Conf. Proceeding Ser.*, 2014.

[81] S. Sreejesh, S. Mohapatra, and M. R. Anusree, “Business research methods: An applied orientation,” *Bus. Res. Methods An Appl. Orientat.*, pp. 1–281, 2014.

[82] B. Ayyub, “A practical guide on conducting expert-opinion elicitation of probabilities and consequences for corps facilities,” *Inst. Water Resour. Alexandria, VA, USA*, no. January, 2001.

[83] Z. Wang, “Understanding and managing the challenges of distributed scrum teams,” no. September, 2020.

[84] F. Anwer, S. Aftab, S. Shah Muhammad Shah, U. Waheed, S. M. Shah, and U. Waheed, “Comparative analysis of two popular agile process models: extreme programming and scrum,” *Int. J. Comput. Sci. Telecommun.*, vol. 8, no. 2, pp. 1–7, 2017.

[85] M. A. Akbar, J. Sang, Nasrullah, A. A. Khan, M. Shafiq, and Fazal-E-Amin, “Towards the Guidelines for Requirements Change Management in Global Software Development: Client-Vendor Perspective,” *IEEE Access*, vol. 7, pp. 76985–77007, 2019.

[86] S. Dorairaj, J. Noble, and P. Malik, “Effective communication in distributed agile software development teams,” *Lect. Notes Bus. Inf. Process.*, vol. 77 LNBIP, pp. 102–116, 2011.

[87] M. A. Razzak and R. Ahmed, “Knowledge sharing in distributed agile projects: Techniques, strategies and challenges,” *2014 Fed. Conf. Comput. Sci. Inf. Syst. FedCSIS 2014*, vol. 2, pp. 1431–1440, 2014.

[88] F. Q. B. Da Silva, C. Costa, A. C. C. França, and R. Prikladinicki, “Challenges and solutions in Distributed Software Development Project Management: A systematic literature review,” *Proc. - 5th Int. Conf. Glob. Softw. Eng. ICGSE 2010*, pp. 87–96, 2010.

[89] V. Lalsing, “People Factors in Agile Software Development and Project Management,” *Int. J. Softw. Eng. Appl.*, vol. 3, no. 1, pp. 117–137, 2012.

[90] M. Hossain, "Scrum Practice Mitigation of Coordination Challenges in Global Software Development Projects: An Empirical Study," *Univ. New South Wales Sydney, …*, 2011.

[91] E. Hossain, M. A. Babar, and J. Verner, "Towards a Framework for Using Agile Approaches in Global Software Development," pp. 126–140.

[92] D. Dqg, "[1] D. Dqg, "0G 6Krdle 5Dkpdq Dqg $Ulmlw 'Dv," vol. 668, 2015.0G 6Krdle 5Dkpdq Dqg $Ulmlw 'Dv," vol. 668, 2015.

[93] F. Zieris and S. Salinger, "Doing scrum rather than being Agile: A case study on actual nearshoring practices," *Proc. - IEEE 8th Int. Conf. Glob. Softw. Eng. ICGSE 2013*, pp. 144–153, 2013.

[94] R. Phalnikar, V. S. Deshpande, and S. D. Joshi, "Applying agile principles for distributed software development," *Proc. - Int. Conf. Adv. Comput. Control. ICACC 2009*, pp. 535–539, 2009.

[95] R. Giuffrida and Y. Dittrich, "A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams," *Inf. Softw. Technol.*, vol. 63, pp. 11–30, 2015.

[96] S. Kazi, M. S. Bashir, M. M. Iqbal, Y. Saleem, M. R. J. Qureshi, and S. R. Bashir, "Requirement change management in agile offshore development (RCMAOD)," *Sci. Int.*, vol. 26, no. 1, pp. 131–138, 2014.

[97] M. Hron and N. Obwegeser, "Scrum in Practice: an Overview of Scrum Adaptations," *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, pp. 5445–5454, 2018.

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

[99] T. Kamal, Q. Zhang, M. A. Akbar, M. Shafiq, A. Gumaei, and A. Alsanad, "Identification and Prioritization of Agile Requirements Change Management Success Factors in the Domain of Global Software Development," *IEEE Access*, vol. 8, pp. 44714–44726, 2020.

[100] M. A. Akbar *et al.*, "Success factors influencing requirements change management process in global software development," *J. Comput. Lang.*, vol. p.

112–130, 2019.

[101] 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,” *Proc. - 2012 IEEE 7th Int. Conf. Glob. Softw. Eng. ICGSE 2012*, pp. 154–158, 2012.