Master of Science in Software Engineering
September 2020

Understanding and managing the challenges of distributed scrum teams

September 2020

LUJIE WU
ZIYUE WANG

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s):
LUJIE WU
E-mail: luau18@student.bth.se

ZIYUE WANG
E-mail: ziaa18@student.bth.se

University advisor:
Dr. Kwabena Ebo Bennin
Department:Blekinge Institute of Technology (BTH), SERL-Sweden

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

ABSTRACT

Context. Distributed software development becomes increasingly common with the trend of globalization. Scrum, as one of the methods to realize agile, is gradually accepted by more and more people and applied to actual industrial production. Although there have been some successful cases of distributed Scrum team development, the description of these successful experiences may not be exhaustive enough and not applicable to all teams. There is still a great demand for actual industrial case studies, especially related research on specific teams and detailed challenges encountered. In order to enable more distributed scrum teams to better handle various challenges in the development process, further research on the challenges the teams encounter and how to solve, mitigate, and avoid the impact of these challenges is necessary.

Objectives. The main objectives of this research are to investigate the challenges faced by distributed Scrum teams in the development process, the factors that cause them, and how to deal with these challenges.

Methods. We conducted a systematic literature review and obtained the most common problems encountered by distributed Scrum teams and parts of the factors that caused this problem. On this basis, we conducted case studies on 2 large companies in Asia. We used archived data to know the basic information of the case team and a semi-structured interview was used to understand the problems they encountered and their opinions.

Results. During this study we found that the most common challenge encountered by distributed Scrum teams was "Communication among stakeholders". Totally 16 factors were found that could cause this challenge. The two main factors were “Team members have insufficient knowledge or different skill levels” and “Are not familiar with each other or have differences”. And 42 solutions were provided after we integrated the information obtained.

Conclusions. We conclude that communication is a matter of great concern, whether in the literature we used or in our case team. The factors and solutions given are only for reference by teams of similar types and development backgrounds. Further researches on other different types of teams and other challenges encountered are also necessary.

Keywords: distributed, scrum, software, development, challenges, factors, solutions

ACKNOWLEDGEMENTS

First of all, we would like to thank our supervisor, Dr. Kwabena Ebo Bennin, for giving us continuous guidance and encouragement during the research process. He led us through all stages of writing this article. Without his consistent and illuminating instruction, this thesis could not have reached its present form.

Secondly, we would like to thank Professor Tony Gorschek and Professor Michael Unterkalmsteiner for giving us precious inspiration in our research field, and providing continuous support in this research project.

Finally, we would like to thank each interviewee of our project for their participation and cooperation. And last, but not least, we would like to give a special thanks to our family and friends who supported us through the entire process and encouraged to strive for our goal.

CONTENTS

1 INTRODUCTION

Kaur et al.[58] mentioned that due to the globalization of many organizations, the increasingly complex and competitive market situation, Global Software Engineering (GSE) is becoming popular in today's industries. They defined Global Software Engineering(GSE) as "Software development with teams situated at different geographic locations, from different national and organizational cultures, and different time zones". Such kind of development is known as Distributed Software Development(DSD), Global Software Development (GSD).

According to Farooq et al.[83], the current trend in the software development industry was to move towards global software development since it could provide software companies with certain advantages, such as access to skills and cheap labor, lower development costs, and others. But at the same time, global distributed software projects also faced many challenges and issues, like distance and culture. In order to minimize the negative impact of these challenges and issues, agile software development was widely adopted and accepted for its flexible methods to manage requirement volatility and emphasizing cooperation between customers and developers. Scrum, as a popular and widely used lightweight framework for Agile, there was growing interest in applying it to global software development projects. Hossain et al.[7] said it had been proved that Scrum was very suitable for global software development projects and was indeed an effective way to manage projects with many small, juxtaposed development teams.

Through reading the relevant literature, on the one hand, we found that Paasivaara et al.[9] mentioned that there had been some successful projects developed by distributed scrum teams, and Ghosh[4] also mentioned that these distributed scrum teams encountered many problems and challenges in the development process, but these literature rarely mentioned the experience of using scrum framework in distributed projects. On the other hand, Akif et al. [117] mentioned that with the advancement of technology and the improvement of tools, the types of problems encountered by distributed Scrum teams were also constantly changing. In addition, Paasivaara et al.[80] noted that the literature on real industrial case studies reporting on experiences of using agile methods in distributed projects was still scarce. In order to enable more distributed scrum teams to better handle various challenges in the development process, our research problem is to identify the challenges encountered by the distributed scrum teams, the factors that cause these challenges and the solutions used to solve, mitigate, and avoid the impact of these challenges.

We got the most common issue encountered by distributed scrum teams through a systematic literature review and found parts of the factors that caused this most common issue in the literatures. Then we conducted case study on a company, through archiving data and interview to understand the distributed Scrum teams' views on the causes of this most common issue and how they dealt with the issue. Through the analysis and summary of interviewees' answers, we provided solutions for the most common issues.

Through reading various literatures, we found that people still had strong interests in the challenges encountered by distributed Scrum teams and how to solve these challenges. Besides, there was still a great demand for articles on research on specific industrial cases, especially articles that conducted in-depth research on a specific issue in a specific case team.
Through our analysis of the results obtained by the research methods, we observed "communication among stakeholders" was the most common issue. In our research, we found and sorted out totally 16 factors that could cause this issue and 42 solution could be used to deal with the issue related to communication.

These solutions provided a reference for those distributed scrum teams with similar background and personnel distribution to our team and we thought software practitioners who worked in distributed environments should consider improving communication.

Outline

The following chapters' topics are described as follows.

In Chapter 2, we summarize the history and background of distributed Scrum to date, as well as some of the related work related to challenges encountered in distributed.

In Chapter 3, our research problem, our two experimental methods (SLR and Case Study, respectively) and our research motivation are summarized. And in this section we describe the study steps in detail.

Our conclusions will be discussed in detail in Chapter 4 and 5. In Chapter 5, we also address the threat to effectiveness. In Chapter 6, we summarize all our studies and put forward further work.

2 BACKGROUND

In the following part, we introduce some information about distributed scrum software development and our research topic through background and related work.

2.1.1 Distributed software development

According to Muhammad et al.[49], in the past, software development was considered as an internal affair. The companies had their own development sites where the collocated teams developed software. Shrivastava et al.[61] mentioned that in recent years, the software market situation has become increasingly complex and competition in software development environment had intensified.

As mentioned by Al-Zaidi et al.[46], in order to reduce time to market, improve product's quality, achieve round-the-clock development, access to cheaper skilled resources, obtain local knowledge and grow productivity[46], many organizations had started developing software remotely and the increasing investments enabled a move from local to global markets based on Prikladnicki et al.[63] According to Kaur, et al.[58],Global Software Engineering (GSE) was becoming popular in today's industry due to the globalization of many organizations.

According to Kaur, et al.[58], GSE could be defined as "Software development with teams situated at different geographic locations, from different national and organizational cultures, and different time zones". Such kind of development was known as Distributed Software Development(DSD) and Global Software Development (GSD). Prikladnicki et al.[64] thought that the characteristic of DSD was that software project teams were geographically distributed. Carmel et al.[62] mentioned that when distance became global, this was the hallmark of GSD.
According to Layman et al.[65], DSD ranged from team members being distributed over adjacent buildings to being distributed over different continents and GSD was the special case of DSD in which the dispersion of the team extends across national boundaries as mentioned by Sahay[66]. So in our research, we considered both as DSD. The definition of DSD was described as following based on Wohlin al[67] and Ghani[48]: members from different nationalities, different cultural backgrounds, different geographical locations and potential time zones form a distributed team, and they can develop remotely in different parts of the software project (independent tasks) with or without any face-to-face interactions. According to Ghani[48], there were three types of distributed teams and their team members:

Single-team and distributed team members: It refers to a single team of which members are distributed in different physical locations. As described in Anand[71], the members of a team were distributed across two European countries and India.

Multiple distributed teams and co-located team members: It refers to multiple teams that are located in different physical locations, and the team members of the same team are located in the same place. As mentioned by Mohan[70], a software was developed with 16 teams distributed across three continents (Asia, Europe and North America). And all team members belonging to a team were in the same location.

Multiple distributed teams and distributed team members: It means that the teams are located in different locations, and the team members belonging to the same team are not in the same place, but are distributed in various places. According to Gupta et al.[72], the teams and team members within the teams were distributed across 3 countries (Germany, India and USA).

The Figure 1 below illustrates these 3 types of distributed teams. The people image represents a single team member. The same color of the people's images means that these people are distributed in the same team. And the oval encircling the people's image represents the geographical location of the team. Different colored ovals represent different geographical locations.

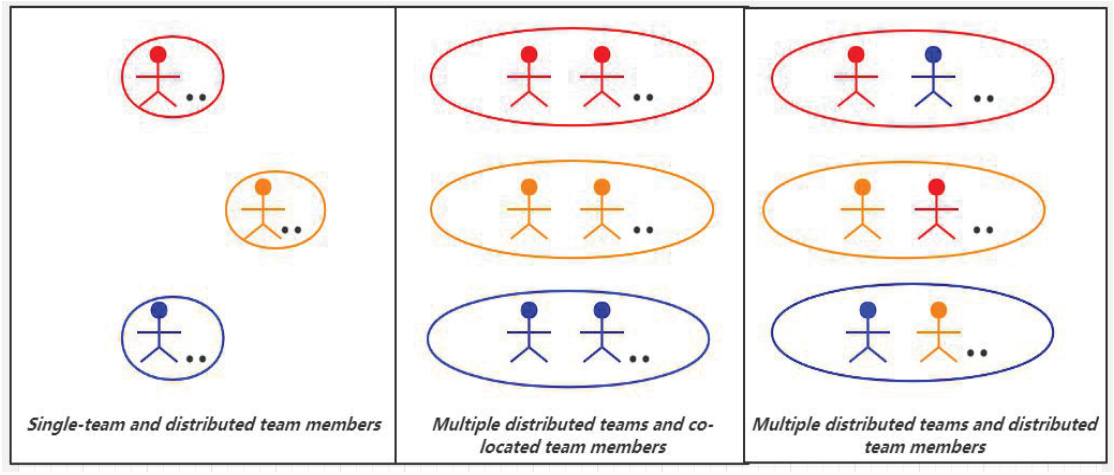

Figure 1.Three distributed scrum organization charts

According to Kaur[58], Rahman[50] and Vasudeva et al.[51], although distributed software project teams helped in cost reduction, improving quality and increasing productivity, it also brought a lot of challenges, like communication, coordination and control,etc. And these all challenges mainly arose from three kinds of distances which could be considered as root factors: temporal distance, geographical distance and cultural distance. Based on Holmström et al.[68], the combination of all these distances made distributed software development more complex. In our study, we considered them as DSD properties. The details of these three distances(DSD properties) are explained as follows according to Agerfalk et al.[69]:

Table 1: DSD properties

1.Temporal distance: It refers to a directional measure of the time misalignment experienced by two actors who want to interact. Temporal distance is mainly caused by time zone difference or time shifting work patterns. We should pay attention to the temporal overlap of parties, which can facilitate communication, and temporal coverage.Generally speaking, a smaller time distance can increase the chance of synchronous communication in time, but may reduce management options.
2.Geographical distance: It refers to a directional measure of the effort required for one actor to visit another at the latter's home site. It is best measured in ease of relocating rather than in kilometres. Two places in the same country with direct air links and regular flights between them can be considered close although they are far apart. And two places can be considered far although they are geographically close if the transportation infrastructure between them is inconvenient or there is an intermediate boundary. So generally, low geographical distance provides more scope for periods of co-located, inter-team working.
3.Cultural distance: a directional measure of an actor's understanding of another actor's values and normative practices. It has many facets, including organisational culture, national culture and language, politics, and individual motivations and work ethics. It is possible to have a low socio-cultural distance between two actors from different national and cultural backgrounds who share a common organisational culture, but a high distance between two co-nationals from very different company backgrounds. And In general, low socio-cultural distance improves communication and lowers risk.

2.1.2 Agile Software Development

As mentioned by Vasudeva et al.[51], agile software development referred to a set of software development methods which aimed at more flexible and lighter development processes, making them more responsive to changes. According to Kaur et al.[58], in the past ten years, the introduction of agile software development had brought great changes to the field of software engineering. Williams et al.[73] mentioned that agile methods were gaining popularity in highly uncertain software

development environments. The characteristics of agile software development process could be summarized as iterative and incremental development, people-oriented development, customer collaboration. Based on Miller's view[74], light and fast development cycle, adaptability throughout the software development life cycle, time-bound, simple and fast delivery.

According to Pernille et al.[75], agile and distributed software development were two trends that continued to grow rapidly in the software industry today. Many businesses and organizations were trying to combine them to gain the benefits of both, but the inherent challenges of this combination often led to serious complications that could jeopardize the successful completion of a software project.

The root cause of the challenges in distributed agile development was the opposite of the conditions required by the two approaches. It could be said that the main purpose of agile methodology was to focus on the communication between the people involved in the development of software systems. According to Beck et al.[76], as the manifesto of agile software development states in the list of principles: "The most effective and effective way to communicate information to and within the development team is face-to-face conversation" . Therefore, the inevitable characteristics of distributed development (such as temporal, geographical and cultural distance) obviously prevented the development team from fully complying with the defined principles of agile software development, making distributed agile development a higher risk of failure.

2.1.3 Scrum

Erickson et al.[78] mentioned that scrum was an iterative and incremental agile project management method. At this stage, people were increasingly interested in researching the practice of Scrum for DSD. Agile methods could reduce development time, shorten response time, improve quality, reduce development costs, and improve communication between teams and participants.

According to Dullemond[47] and Erickson et al.[78], we knew that the basic configurations of Scrum were as follows:
1: Role assignment
Scrum development involves three roles: Product Owner, Scrum Master and Development Team
2. Four meetings
The four meetings refer to the Sprint planning meeting, daily meeting, Sprint review meeting, and Sprint Retrospective.
The Scrum flowchart is shown as below:

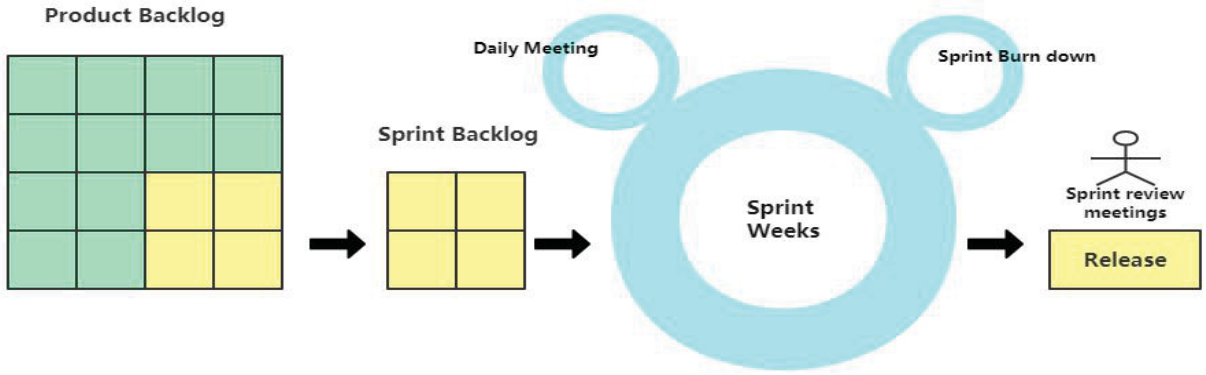

Figure 2.Scrum flowchart

2.1.4 Scrum in distributed software development

In the recent years, due to hot progression in technologies that shaped the businesses across the boundaries, the doors of a distributed software development was opened as Prikladnicki et al.[64] said. Rahman[50] and Cristal et al.[11] mentioned that distributed software development had become popular and increasing in the software industry. As we mentioned above, DSD brought many potential benefits, but also brought lots of challenges like communication, coordination, decreased visibility

into project status and configuration management, etc. based on ÅGERFALK and Phalnikar et al.[79][77]

According to Paasivaara et al. [10], many existing countermeasures were based on the assumption of stable requirements to develop a clear modular structure to reduce communication traffic. While many distributed software development teams needed to face situations like dynamic business environments as Lassenius et al. said[80], unstable demand and uncertain implementation technology based on Paasivaara et al.[10], etc. Due to agile method was especially appropriate for projects facing high uncertainty as said by Cockburn and Highsmith[81], suitable for dealing with uncertainty and changing requirements according to Ghani et al.[48] and was useful in distributed projects to reduce the negative effects of distance on communication, coordination and control based on Carmel and Agarwal[82], more and more software development organizations had attempted to combine the benefits of agile methods and DSD according to ÅGERFALK and Farooq et al.[79][83] According to Shrivastava and Modi et al.[84][61], agile software development referred to a set of software development methodology which aimed to make the development process more flexible by avoiding pre design, so that they were more sensitive to change. The adoption of agile software development processes had increased over the years mentioned by Diebold et al.[85] and they did help to overcome some challenges faced by distributed development according to Smits and Pshigoda et al.[86]. Sriram and Mathew[87] also mentioned that Scrum was the most widely used and most preferable agile method that suited DSD. In addition, Hossain and Paasivaara et al.[19][5] also mentioned that scrum could be effectively used in GSD to minimize distributed challenges and scrum in distributed teams was beneficial for project's quality and performance.

According to Sriram and Mathew[87], although combining scrum with DSD could produce amazing results, while as an agile development process framework, scrum was strongly driven by immediate and direct communication, and emphasized people to people collaboration - usually more effective for smaller teams in the same location. On the other hand, distributed software engineering required many rules and forms to coordinate teams in different locations as said by Lous et al.[6], so some new challenges and risks would also be brought if they were combined based on Shrivastava and Date[51].

According to Modi and Alsahli et al.[84][53], there had been a gradual stream of literatures on the usage of agile methods in distributed settings which illustrated the challenges in this area and lots of literatures revealed many agile practices could be used to mitigate some DSD challenges. But most of the literatures only gave rough descriptions of the various challenges or solutions mentioned, but did not go deep into a certain challenge and its corresponding solution, and some of them were not mainly about scrum. Although some reports stated the successful implementation of agile methods in distributed software development[58], they rarely mentioned the detailed experience of using scrum framework in distributed projects. Besides, the literature on real industrial case studies reporting on experiences of using scrum in distributed projects was still scarce. So our goal is to study the most common challenges faced by distributed scrum teams, the resulting factors and propose reasonable solutions based on the experience of real industry cases.

2.2 RELATED WORK

Through the analysis results introduced in the previous section, we had a rough idea of the concept and development trend of distributed Scrum. We knew that the use of agile methods in distributed teams was still developing at a high speed, and scrum was still the mainstream method in agile development. While research on the challenges faced by scrum teams used for distributed software development in practical cases was still scarce.

Durasiewicz et al.[10] introduced the findings of a multi case study of Scrum practices from 19 members of two small and one medium-sized distributed scrum projects. The purpose was to find out how scrum could be successfully applied to distributed development. The author introduced the workflow of Scrum method used in these cases and analyzed the advantages and challenges of these

processes in distributed development which we very much approved of. Then the author provided the practices used by the case projects to support distributed scrum. Although the explanation of Scrum workflow in this article was very detailed, we thought there might be some improvements in presenting the challenges faced by the team in the case and the corresponding solutions. We could not quickly determine the specific challenges and solutions faced by the case studied in this article.

Paasivaara et al.[5] mentioned that combining large-scale global software development with agile still faces many challenges. The article reported a case study of the application of Scrum practices in a 40-person development organization distributed between Norway and Malaysia.The analysis results showed that through unofficial distributed meetings, centralized version control and other practices could indeed have a positive impact on the success of the project, but still faced some challenges: there was no possibility of video conferencing, the silence caused by misunderstanding requirements and distance. Besides, the author mentioned that during the research process, only seven people could accept face-to-face or telephone interviews. For those interviewees who couldn't meet, it was particularly difficult to establish a confidentiality atmosphere, which might lead to the inability to say everything and we should consider how to allow interviewees to provide as much information as possible in our research. Due to the project process of the case provided in this article was unique, we thought that this article was more focused on providing the experience of distributed projects in these cases and didn't not give too much description about the challenges and corresponding solutions.

The Lautert et al.[39] conducted a survey of a large distributed Scrum team to determine the challenges that the distributed environment might bring to the Scrum team. And through the results of exploratory analysis we could know: the interviewed professionals had the knowledge level of Scrum practice, communication was one of the main challenges in a distributed environment, and the interviewees would prefer phone calls when considering the type of communication. This article investigated the specific aspect of communication, we thought it was good to set the scope and focus of the research question, which helped discover more detailed information in this area. Besides, Lautert et al.[39] provided lots of charts for data support, which helped present information and facilitate us to read and understand. But the shortcoming was that the data provided in this study accounted for about 19% of the population surveyed, and the rest might provide more relevant and useful information.

According to Marcel et al. [2], the author spanned three student scrum teams across Canada, Finland, etc., working on a large-scale project, and tracking students before and after the course to study their work progress, the degree of adaptation to distributed scrum and challenges encountered in development. The final survey results showed that communication was the most difficult issue they met. There were three points involved here, namely the importance of early communication, the challenges encountered by the communication between the teams, and the difficulties of communicating with customers. 1. All teams use daily Scrum as a video conference twice a week. 2. Choose appropriate communication and coordination software. 3. Actively communicating with PO and other good behaviors would help alleviate the above difficulties. But in this study, students didn't not have rich development experience, and because the team had significant differences in knowledge, often paired programming was chosen between the two teammates. It was difficult to verify whether these tasks are feasible in actual development.

As mentioned by Altaf et al. [18], the author conducted SLR analysis and investigated the influencing factors of agile adaptation in global software development, and finally got two aspects. The first aspect referred to the success factors of agile method adaptation to distributed development (for example, it was more flexible than traditional development, emphasizing communication with customers, etc.) The other was the challenge that agile faces in distributed development (for example: lack of documentation in agile development, unclear role of workgroup, etc.).This article discussed the feasibility of using scrum in distributed teams, and concluded that it is feasible to use scrum in distributed teams. However, we thought that his research scope was relatively small, and most part of the article he studied described the advantages of using agile methods in distributed development, not

our theme——challenges. We wanted to focus more on the challenges faced by distributed scrum teams,and looked for the factors that cause these challenges and the ways to solve them.

Based on Singh et al. [42], the author emphasized the importance of using agile applicability parameters when using agile in distributed development. It was proposed that the development process should be improved through open communication and minor adjustments when facing different situations, such as designing in the early stage and maintaining a public code repository. And the author also had roughly determined the eight factors that might cause difficulties in using Scrum in distributed development. They defined all 43 sub-factors. But most of the articles cited by the authors were from Indian companies and might not be suitable for all situations. In this article, most agile proponents tried to claim agile success in introductory projects, so it might not contain all the difficulties that arose when using scrum in a distributed team. And there was no specific description of the factors causing the challenge, so as to correspond to the solution.

As mentioned by Bjarnason[123], coordination and communication within software development are affected by distances. The effects of geographical, social cultural and temporal distances are fairly well known and researched for distributed software development (DSD).The author believes that distance may be a fundamental factor in the coordination and coordination of software engineering activities. In the author's previous research, 13 distances about distributed development have been sorted out, 8 of which are related to people. The most mature research is about the distance between people, namely, geography, time and society. culture. Based on the previous 8 distances, the author conducted case analysis and summarized 8 abstract practices to alleviate the influence of distance in distributed development. In this article, although an interview is used for case analysis, it does not specifically summarize the detailed risk and challenge. The description of the key issue is abstract and difficult to understand. And the solution may not be targeted.

After analyzing the existing related work, we rarely found literature describing the detailed overview of the surveyed projects, the technology used to analyze the data collected, and the specific details of the development process in the form of data. Most of them were based on the respondents' subjective answers to get abstract results, the contents of these personal statements might lack pertinence and could not cover all aspects of the project development process. And these answers might come from different project teams and had different development environments, so it was difficult to summarize them when analyzing data to make them representative.

Many of the articles were a rough analysis of all the issues encountered in the development of distributed Scrum without focus. And there might be no detailed introduction to the project and team information, that was, the specific development environment and background were not explained. The successful experience provided by such articles and the solutions applied when they encounter challenges might not be of reference to some readers.

In our research, we combined theory with practice. For theory, we used system literature review to find the most common issues encountered by the distributed scrum teams in the development process and parts of the contributing factors. Then in case study, we interviewed the case team to find out more about the causes of the issues found in SLR and summarized solutions to detailed issues by analyzing and summarizing the answers of the interviewees.

We found common challenges through general teams in SLR and conducted interviews in our special case teams. In addition, we learned the project background and personnel information of our case, and also made an in-depth investigation on the important part of the Scrum process—— meetings. In the end, we combined and showed the factors, detailed issue scenarios and solutions found, so that readers could intuitively see the challenges encountered in the development of a distributed scrum team, the factors caused and the corresponding solutions when reading.

3 RESEARCH METHOD

Through the research results of related work in the previous chapter, we had a better understanding of the challenges faced by the distributed scrum team in the development process, and identified the knowledge gap in our work and the scope of our research.

Identification of gap:
Although there have been some successful cases of distributed Scrum team development, the description of these successful experiences may not be exhaustive enough and not applicable to all teams. Many people pay more attention to the specific problems and specific countermeasures encountered in distributed Scrum software development, rather than just the successful experience. There is still a great demand for actual industrial case studies, especially related studies and detailed challenges for specific teams. In order to enable more distributed Scrum teams to better cope with certain specific challenges in the development process, we studied the most common challenges faced by distributed Scrum teams and how to deal with them.

This chapter described the aim of the research, the research objectives, the research questions, and the methods used to achieve the required research objectives and answer the defined research questions.

Aim:
The aim of this research work is to identify and understand the most common issues encountered by distributed scrum teams in the software development process, the corresponding causes, and to propose solutions to these issues.
This aim is achieved by fulfilling the following research objectives:

Research Objectives:
Obj1:Identify the most common issue encountered by distributed Scrum teams in the reviewed literature.
Obj2:Identify the factors that cause this most common issue.
Obj3:Propose solutions for this issue in this context.

Research questions:
RQ1:What is the most common issue encountered by distributed scrum teams?
RQ2:What are the factors that cause this common issue?
RQ3:What are the solutions that can be used to solve this issue?

Motivation:
RQ1 helps us understand the various problems encountered by the distributed Scrum teams during the software development process in different kinds of literatures, so as to get the most common issue(Obj1) and part of corresponding factors that cause this most common issue mentioned in the literature. This points out the direction and scope for our follow-up study of the factors that cause this issue. Then based on the result of RQ1, RQ2 enables us to know the important factors(Obj2) that cause this most common issue, which provides essential information for us to come up with the appropriate solutions next. With the results of RQ1 and RQ2, we can propose the suitable solutions(Obj3) for the most common issues encountered by the distributed Scrum team in the software development process. Therefore, these development teams can be more flexible to cope with or avoid the negative impact of these issues.

Alternative methods:
a)Experiment
For research methods, the first thing we consider is to get the data we want by doing experiments. It is used to measure one variable on another variable in a controlled environment as Runeson et al.

mentioned [118]. And the two important principles of doing experiments are to control variables and make comparisons. The idea is to study the influence of the change of an independent variable on the dependent variable by making comparisons. There are many variables for the object we want to study. Whether the team is distributed and whether the scrum framework is used are two basic variables. We need to ensure that the team is distributed and uses the scrum framework for project development. Secondly, since it is an exploratory study and it is processed in an uncontrolled environment. The scrum experience and knowledge level of each team member will greatly affect the results of the experiment, and we have no idea how to find an appropriate independent variable and control group to carry out the experiment, so we give up using experiments as our research method.

b)Survey
Then we think of Survey. It is always used to derive a conclusion from specific
population. Runeson et al.[118] mentioned that It is descriptive in nature which aims to portray the current situation. In addition, Wohlin et al.[93] thought survey aims at providing broad overviews, the respondents often answer questions based on their subjective opinions and interests, and due to limited time, they probably won't mention the trivial things in actual development, so we may not be able to get some specific details of the project development process. Thus， Survey is not inappropriate for our research work.

c)Action Research
When it comes to Action Research, Paasivaara et al.[80] said it is used to bring
influence on a subject. As our research is proposed in an uncontrolled environment, many uncertain factors may affect our research, Action Research is not suitable.

d)Systematic Literature Review
Through this method, we want to collect the data related to distributed scrum teams and the issues they encountered. This lays the foundation for providing background knowledge related to the issues encountered by distributed Scrum teams in a wide range of contexts, and provides a comparison item for subsequent data comparisons. By browsing the previous literature, we will more effectively conduct inductive comparison and analysis of the data.
There are mainly two approaches to review the previous literature: SLR(System Literature Review) and literature review. According to Nightingale[119], SLR aims at identifying, analysing and interpreting all available evidence related to a specificresearch question. Scientific, rigorous ways of identification, analysis and interpretation are required to achieve this purpose. Kitchenham et al.[114] mentioned that SLR is a methodologically rigorous review of research results. The purpose of SLR is not just to aggregate all existing evidence on a research question; it is also intended to support the development of evidence-based guidelines for practitioners. Then according to Boell et al.[120], literature review is a survey of academic resources related to a specific research topic, with the goal of providing a comprehensive look at what has been said on the topic and by whom. Through Reed[121], literature review can help us provide our readers with background information about our research and also show how familiar we are with our research in the field, and how our work has contributed to the challenge of expanding the knowledge base in the field.
The topic we are studying is about various issues encountered in the development of distributed Scrum teams. On the one hand, although this is a very broad topic covering a wide range of contents, this topic includes specific research areas and questions. First, we must ensure that the development teams of the companies mentioned in the searched literature are distributed, and they should use Scrum which is an agile development method to develop the project. In addition, the literature should also mention content related to the challenges or issues encountered and solutions, not just a general overview or a brief introduction. SLR is usually used to search for papers related to a specific research question, and all relevant research should be found, which is more stringent and consistent with our selection strategy as Wohlin et al.[93] mentioned. On the other hand, our research aims to provide evidence-based guidelines for distributed Scrum practitioners. With SLR, we can collect all existing research questions and analyze them to get the issues that practitioners encounter and the solutions that they think are effective.

This can help these practitioners better reduce, solve or avoid the negative impact of these issues. Through data induction and analysis of these search documents, we are expected to obtain various issues encountered by distributed Scrum development teams, the causes and solutions of these issues. Then we can learn about the most common issues these teams face and what factors cause these most common issues. Therefore, we think SLR fits our theme.

e)Case study
Then we think of case study. Through the study of Wohlin et al.[93], we know that case study refers to the study of a single entity or phenomenon in the real environment within a specific time limit. With the development of software globalization and the maturity of Scrum framework, it is a trend for distributed scrum team to develop software, but this process is complex and dynamic, and has too many variables. The success of distributed scrum team development projects depends on multiple evidence sources. To study the problems encountered by the distributed scrum team in the development process, we need to use the previously recognized theoretical proposition to guide the data collection and analysis, which is Yin's[122] supplement to the characteristics of case study. At the same time, we also consider that case study does not need to strictly control some things like experiments. Our research topic belongs to the field of software engineering. Too many factors will affect the output results of software engineering activities. We cannot strictly divide the distributed scrum team and their environment. Therefore, we believe that case study is also suitable for our theme research method.

Overview of Research Methods
First, we obtained relevant literature through a systematic literature review (SLR). In SLR we divided into two independent processes, one was the collection of data, and the other was the extraction and analysis of data. After we analyzed the data in the literature, we obtained Obj1.

Then, since obtaining Obj1 from the SLR, we returned to the literature find part of Obj2 for Obj1.Then in the case study, we used archived data and semi-structured interviews to get what caused the distributed scrum team to encounter Obj1 and the corresponding solution.Later in chapter 3.2, we explained the case study process in detail. After combining the two methods, we got all parts of Obj2.

Finally, through our comprehensive analysis and collation of the case study results and Obj2, we got Obj3.

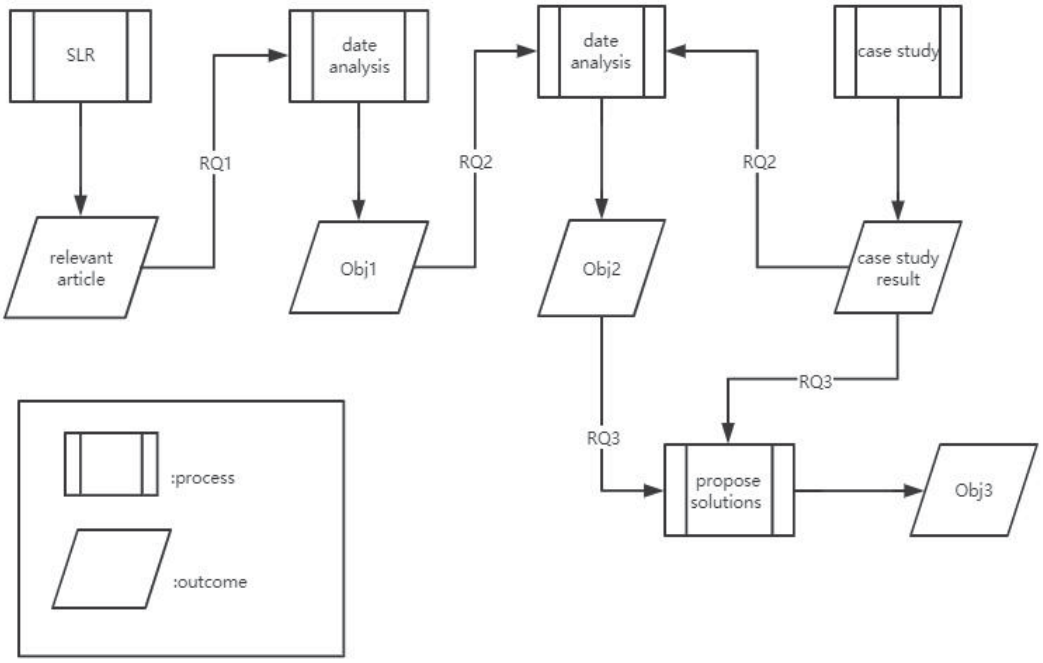

Figure 3.Overview of the research process and selected methods

As mentioned in the Figure 3 and text above, we choose Systematic literature review(SLR) and case study as our research methods to obtain the research objectives, answer the research questions and achieve the aim of our research.

3.1 Systematic Literature Review

The following are the detailed designs and process of our SLR process.

3.1.1 SLR process

Preliminary literature research had shown that there was still a knowledge gap in specific research areas. In order to fill the gap, get the defined research objectives and answer the defined research questions, we performed a systematic literature review. This could help us further understand the relevant knowledge of our research field, obtain Obj1 and gain the answer of RQ1, so as to lay the foundation for the study of subsequent factors and solutions.In this section, we described the specifics of the design and implementation of the SLR in both text and graphics.

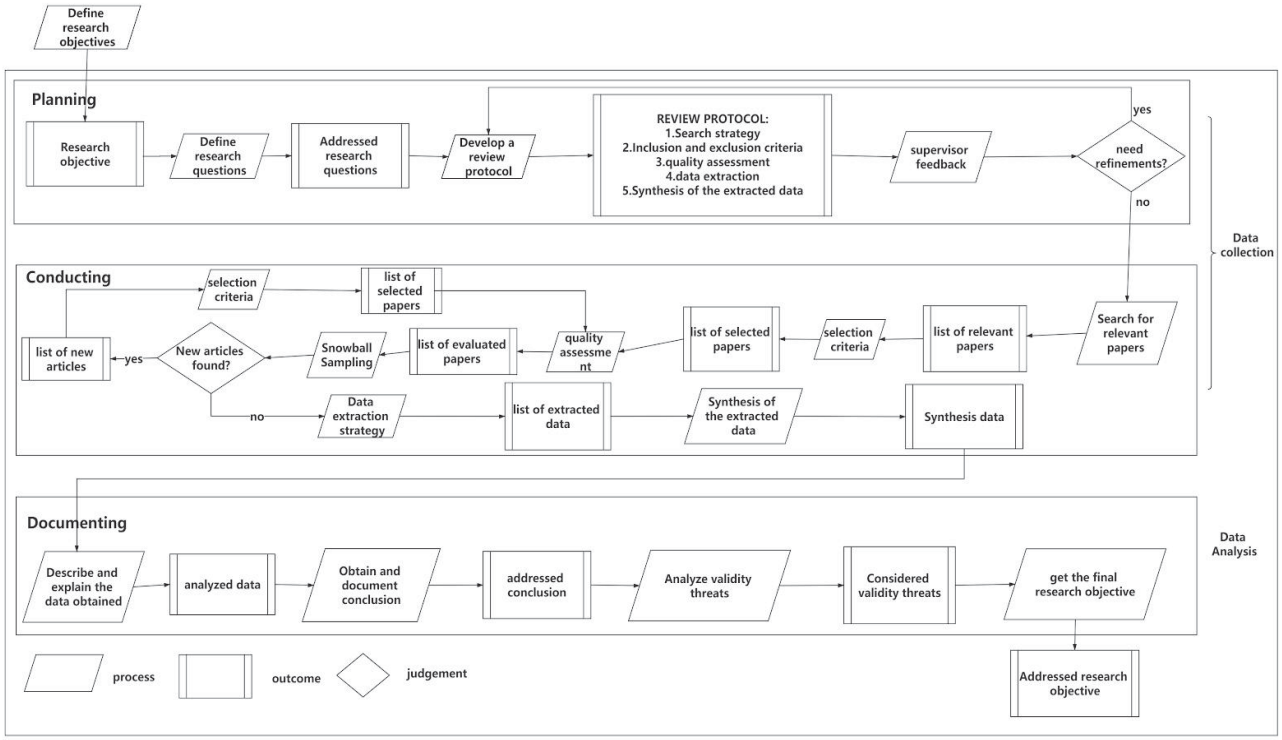

Figure 4.SLR process overview

The SLR process is divided into three main phases according to Kitchenham's[88] guidelines: Planning the Review, Conducting the Review, Documenting the Review. The stages of planning and conducting belong to data collection, and the stage of documenting belongs to data analysis.

After defining the research objective (Obj1) we needed to find, we started the data collection phase of SLR. First in the planning stage, we mainly needed to define research questions(which we had already done in the above) and develop the review protocol. When we identified our research question, then we developed the review protocol, it encompassed search strategy, inclusion and exclusion criteria, quality assessments, data extraction and synthesis of the extracted data.In order to assess the reasonableness of the review agreement we developed, we consulted our supervisors for suggestions. After obtaining the agreement of supervisor, we started conducting the SLR.

In the conducting stage, we used our search strategy to search the papers that were related to our research topic. Then according to our selection criteria, we excluded or included the papers we

searched. Next we assessed the quality of the papers to obtain the list of evaluated papers based on quality assessment checklists. We got the articles that had been filtered by selection criteria and quality assessment so far. Then we also used forward and backward snowball sampling(see the details in section 3.1.5) to get as many relevant articles as possible, which made up for the deficiency of search strategy and mitigated the threat of missing important research. The advantage of using snowball sampling after obtaining primary articles by using selection criteria and quality assessment was it helped mitigate all unrelated articles and avoid alienation of our research topics.We constantly performed the snowball sampling, filtered the papers by using inclusion/exclusion criteria and evaluated the quality of the papers found until we could not find new articles any more or went out of our research scope. The following step was to extract and synthesis data from these screened and evaluated papers by reading them carefully.

Finally, it was the phase of data analysis which encompassed the documenting stage. We had obtained the data we needed, and in this stage, what we first described and explained these data in easy-to-understand expressions and induction ways. Besides, we needed to analyze validity threats of the result and process of SLR from multiple perspectives(see the details in section 5.3.1). In the end, we obtained the addressed research objectives(Obj1) which was the premise for the following research.

The aim of the SLR process is to achieve the Obj1 and answer the RQ1. This section is an overview of the SLR process and a rough description of the steps involved. And the following sections introduce the details of every step.

3.1.2 Search Strategy

As shown in Figure 5 below, we mainly have 4 processes: identifying keywords, selecting resources, creating search strings and performing search.

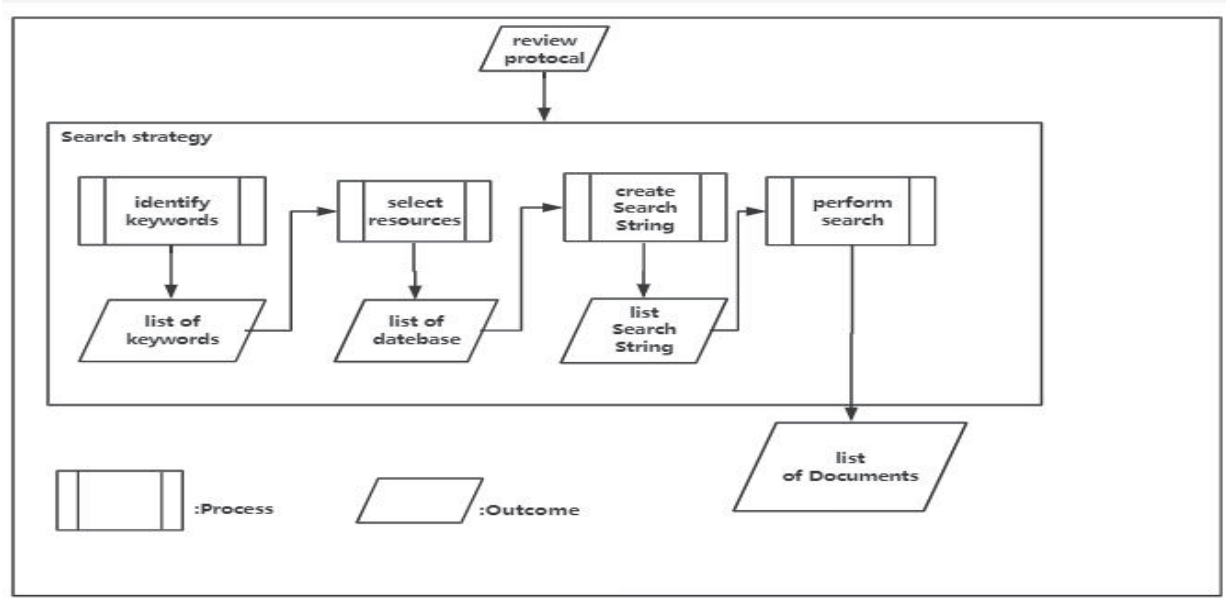

Figure 5.Overview of the search process

We started with defining keywords. It played a key role in getting the papers that were relevant to our research topic. We could preliminarily extract three terms used to combine keywords from our research topic: distributed, scrum and challenge. While our overall idea is to search for as many articles as possible related to our research topic, and then through our screening strategies and quality assessments, we can finally get articles that meet our requirements. So in order to better define our keywords and form the search strings with extensive literature coverage, we referenced Rumsey [89] and Webster's guide[90]. The principle we used here was the keywords should be supplemented with synonyms or synonym groups and broader or narrower terms. So in addition to using “distributed”, we

also added “global”(synonym, as explained in Chapter 2),“distributed software development” (narrower term) and etc.

Furthermore, when we studied the related work, we found that although the theme of many articles was to study the challenges encountered by distributed agile teams in the software development process, their researches were based on the Scrum method or the agile mentioned in their articles referred to the Scrum framework. So in addition to using “scrum”, we also added “agile”(broader term), “scrum method”(synonym group) and etc.

The keywords were chosen based on three core concepts: distributed software development, usage of scrum method and challenge as shown in Table 2 below.

Table 2.Search keywords

Distributed software development	distributed software development, distributed programming, global software development, global software engineering,multi-site software development, distributed software engineering
Usage of scrum method	Agile, Agile method, Scrum ,Scrum method,
Challenge	risk, issue, challenge,problem

Then we explained in detail how the keyword we selected made up the search strings and how search resources were determined. We created a search string based on the identified keywords using truncation, wildcards, and Boolean connectors. In each group of similar words, their Boolean connection is OR, and the Boolean connection between different groups is AND. These similar phrases were finally determined based on our analysis of the literature in the chapter of related work, which clarified the scope and objects of our study.

And we selected five online databases as our search resources. These databases have accumulated peer-reviewed articles and focused on engineering and computer science. We use these resources for major literature searches.

The specific information of search strings and search resources are listed in table 3.

Table 3. Search strings and Search resources

Search strings	(“distributed software development” OR “distributed programming” OR “global software development” OR “global software engineering” OR “multi-site software development” OR “distributed software engineering”) AND (“Agile” OR “Agile method” OR “Scrum”OR“ Scrum method ”)AND(“risk” OR “issue” OR “challenge” OR “problem”)
Digital databases(Search resources)	(1)IEEEXplore (www.ieeexplore.ieee.org/Xplore/) (2)Scopus (www-scopus-com.miman.bib.bth.se/) (3)ACM Digital Library (www.portal.acm.org/dl.cfm) (4)SpringerLink(www.springerlink.com) (5)Wiley InterScience (www.interscience.wiley.com/)

3.1.3 Inclusion and exclusion criteria

We searched lots of articles through using the search strings mentioned above. Then the aim of this step is to find the articles that are valuable for our research.

So we needed to judge which articles were closely relevant to our research topic and could be fully browsed by us. We referred inclusion and exclusion criterias proposed by Khan et al.[91] to filter the articles obtained for primary study.

Inclusion criterias:
I1.The article should be peer-reviewed.
I2.The article should be conducted in English.
I3.The full text can be viewed.
I4.The article mentions the challenges encountered in the development of distributed scrum teams.
I5.The article must be a journal, conference, magazine or book chapters.

Exclusion criterias:
E1.Articles are not in English.
E2.The articles are not peer-reviewed.
E3.Full text is not viewable.
E4.The articles do not mention the challenges encountered in the development of distributed scrum teams.
E5.Articles other than journal, conference, magazine and book chapters.
E6.Duplicity in the articles.

3.1.4 Quality assessment

After getting the list of the papers which were analyzed using the inclusion and exclusion selection criterias, we needed to assess the quality of these papers.
According to Keele et al.[92], in addition to general selection criteria, quality assessment is also important which can provide still more concrete inclusion/exclusion criteria. Wohlin et al.[93] mentioned that by assessing the quality of the primary studies could help us weigh the importance of individual studies when synthesizing results. The quality evaluation of the screened articles showed that the articles we had were reliable and their data could be used, which provided a guarantee for our next data extraction. Therefore, we used quality assessment to further analyze whether the list of selected papers after using selection criterias were reliable or rigorous so as to provide a guarantee for our next data extraction process.

For the papers we got, we comprehensively analyzed the quality of the article from the aspects of rigor and relevance according to the criteria defined in [94]'s study.

The rigor can be considered as the extent of precision and exactness of the research method description used in the particular study. It consists of three aspects: the extent to which context, study design and validity are described. During the assessment these three aspects are scored with three score levels – "weak", "medium", "strong", and the corresponding scores are 0, 0.5 and 1.

As for relevance, it refers to the extent of the industrial relevance by the means of
described realism of the performed evaluation. It consists of four aspects: subjects of the evaluation, level of context description, scale of the performed evaluation, research method used for evaluation. These four aspects are scored with "true" or "false" values, and the corresponding scores are 1 and 0.

Through comparing the detailed descriptions, we found that there were significant differences between the rigor and relevance dimensions. In addition, the ability of the bubble chart used by Munir et al. [95] in the study to quantify and visualize data from several different dimensions also led us to select this quality assessment framework.

In order to facilitate our statistics and analysis of the data obtained from the quality assessment, we designed the specific form to record the results.
The table 4 below is an example of the quality assessment form we used.

Table 4.An example of the quality assessment form

Ref.	Context	Design	Validity	Rigor	Subjects	Context	Scale	Research method	Relevance
[1]	0.5	1	0.5	2	1	1	1	0	3

The bubble chart in section 4.1.2 showed the specific result of assessment of rigor and relevance. Please refer to the appendix for detailed scoring of the quality assessment of each article(including the articles that were ultimately used to extract the data, and those that were not used after performing quality assessment).

3.1.5 Snowball sampling

The snowball sampling is based on Webster and Watson [90] ideas proposed in information systems. It can be applied to many areas of our lives. When it is used as a literature method, its advantage is to drive further study by starting from some relevant papers. After identifying the articles by our selection criterias and quality assessments, we used snowball sampling to discover more potential articles in literature which were relevant to our research topic so as to reduce the threat of missing out on important research and increase the quality and effectiveness of the overall search process.

According to Wohlin's[96] guide, we designed the snowball sampling process as shown in the figure 6 below.

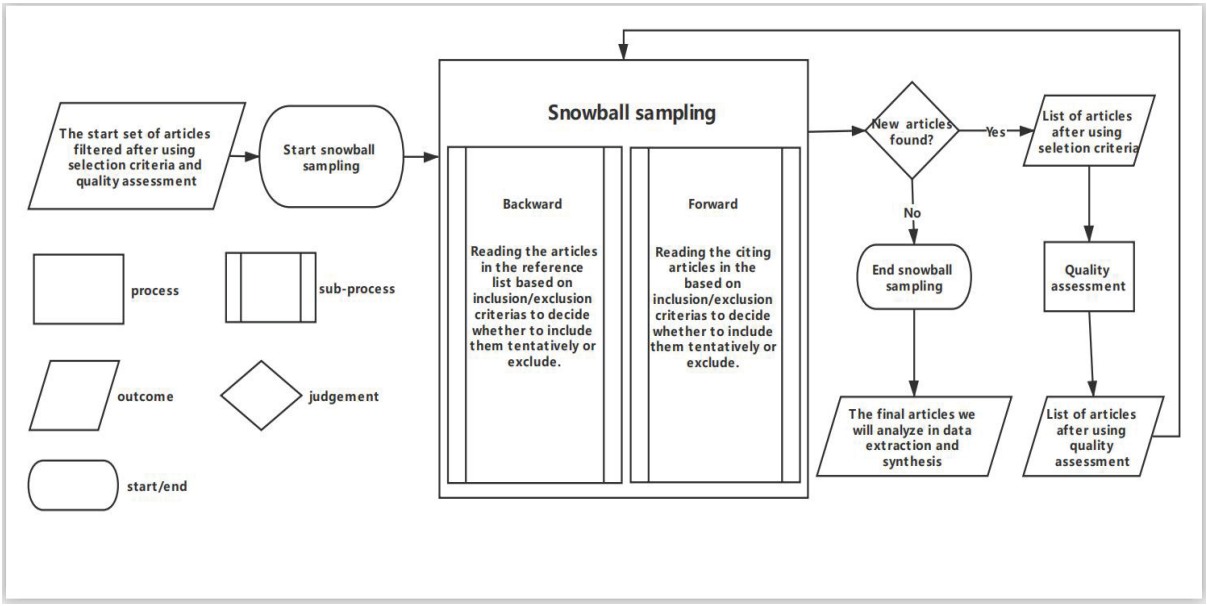

Figure 6.Snowball sampling process

With the set of articles filtered after using selection criteria and quality assessment, we began the snowball sampling.

The snowball sampling is divided into two parts for each article: backward and forward. In backward and forward, we read the articles in the reference list and the cited articles respectively based on inclusion/exclusion criteria to decide whether to include or exclude them tentatively. When we got the list of articles after using selection criteria, we assessed the quality of them. And we continued to perform snowball sampling of the articles included after quality assessment following the above process until there were no new articles found. Then the snowball sampling process ended and we finally got all the articles that we would use in data extraction and synthesis.

3.1.6 Data extraction and synthesis

After quality assessment, we got the articles that would finally be used to extract the data. Since what we needed to find was the most common issue that distributed scrum teams encountered. We began with reading the articles carefully and documented the challenges mentioned in every article. Then we

counted the total number of times each challenge was mentioned in the articles. Through comparing the frequency of all challenges, we could get the most common issue encountered by distributed scrum teams in the literatures. Next we reviewed the articles that mentioned the most common issue and extracted all the factors that caused this issue, which was part of Obj2 and partly answered the RQ2.

Through data extraction and synthesis, we could get the most common issue and the factors that caused it from literatures. It gives us important basic information and also indicates the direction and scope for the follow-up research.

3.2 Case Study

The following are the detailed designs and process of our case study.

3.2.1 Introduction

a)Company background：

The object of our case study is a company registered in 2012, called JARY NETWORK TECHNOLOGY (HONGKONG) LIMITED (http://ww.jary.hk/). Its business scope spans Shandong Province, China and Hong Kong, China, and is composed of three business sectors, involving technology, finance, and commercial real estate. We researched its two subsidiaries, one of which was registered in 2018 under the name of Shandong Linyi Xiangrui, Pawn Co., Ltd. Another company registered in 2017 and located in Hong Kong, JARY Asset Trading Co., Ltd.

b)Project Background

In 2019, companies in the two regions jointly developed an app called DangDuoDuo. The main functions were the sale of financial products, insurance, pawns and financing with Chinese companies. The core business was the auction of pawn corporate bills and financing of companies. Although their subsidiaries were medium in size and had a short establishment time, the management staff had been in the industry for a long time and had extensive experience in agile development.The entire team was divided into two groups by region, with 5 people in Hong Kong and 8 people in Shandong Province. Both groups had their own Scrum Master but only had one Product Owner. There were 13 people in the whole team. When developing applications, they chose the distributed Scrum development model. In this project, because of the financial background, it was necessary to have sufficient financial knowledge, and to tap or find potential customers and recommend suitable financial services, so it involved AI and big data algorithms. The employees in Hong Kong were rich in financial knowledge, mainly responsible for the description of the core business user stories and the development of some financial services, while the Shandong personnels were mainly responsible for the construction of the overall project framework and make some matching algorithms involving AI. Their time was chosen to be an iteration of two weeks, and a total of 3 iterations were experienced.

3.2.2 Case Study Design

The case study was divided into two parts in our study: archived data and interview. Archived data could help us to have a preliminary understanding of project background, project staff and meeting schedule, which provided the necessary basis for our follow-up design of interview questions and processes. Then in the interview, we wanted to know whether the interviewees had encountered the challenges we found in the SLR during the development process, what they thought caused these challenges and how they avoided or responded to them. All of these provide guidance information for us to put forward the solution finally.

3.2.3 Archived data

According to Wohlin et al.[93], archived data usually refers to the organizer's meeting time, organization chart, development documents at different stages, financial records and other previously collected measurement data. Through collecting these data could help us understand some information about the project itself and the people in the organization, although the quality of these data was difficult to assess. But in our study, the quality of these data did not greatly affect our subsequent research because we could further obtain the information we wanted during the communication with the interviewees.

Within the permission of the case company, we got the following three parts of information in the archived data: project background, organization chart and meeting time.Through these information, we could have a general understanding of the background information of the project itself, the staff in this project and the schedule of the meeting. These information could help us better design and formulate interview strategies and questions, and if we found some problems from the information obtained that we could not understand, we could ask questions in a targeted manner and guided the interviewee to recall based on the information, which helped not to miss information and increase the efficiency of interviews.

a)Project background
We wanted to first understand the basic situation of the project, and this might provide us with some help in understanding the content related to the project said by the interviewees more easily in the following interview process.

b)Organization chart
Through the organization chart, we could have a preliminary understanding of the basic information and geographical distribution of the staff in the case, which laid the foundation for the subsequent development of interview strategies and questions.

c)Meeting time
Meeting is an important way for formal communication within and between teams, many information exchanges, process improvements, and strategic decisions take place here. Through viewing the meeting time,we could know how they schedule the various types of meetings in Scrum, how many sprints they had and how long each sprint last.

3.2.4 Interview(semi-structured interviews)

Clifford et al.[97] mentioned that a semi-structured interview was a verbal interchange where the interviewer attempted to elicit information from another person by asking questions. It aimed to help interviewers know what they wanted to find about through Miles et al.[98] The semi-structured interview was great for finding out WHY rather than HOW MANY or HOW MUCH. According to Wohlin et al.[93], semi-structured interview was the most commonly used, because of its flexibility, we were free to determine the order of questions during the interview process, and allowed interviewees to improvise and explore.

Our goal of interview is to to find out whether the interviewees in this case have also encountered the communication challenges we found in SLR, what factors they thought could lead to these challenges, how they avoided or dealt with these challenges, whether they had encountered other communication issues and the corresponding countermeasures, etc. The answers of the interview questions should be detailed, we wanted to know what each interviewee thought and why they thought so rather than simple yes or no, right or wrong. Besides, as a small study, the number of people we wanted to interview was not large. As mentioned by Drever et al.[99], semi-structured interviewing was a very flexible technique for small research which was not suitable for research involving a large number of people. So based on the above reasons, we chose to use a semi-structured interview.

By analyzing the personal views of each interviewee, we were able to summarize various communication challenges, the factors leading to the communication challenges, and the measures taken to solve the challenges, etc. These helped us propose appropriate solutions to various challenges related to communication in a targeted way later.

Through archiving data, we have roughly understood the structure of the project team, the meeting schedule and the completion of the requirements in each iteration.
The detailed results of archiving data is shown in Section 3.2.5.
Based on the obtained information, we designed the following interview questions.

Because these information in archived data were basically provided by the Scrum master from Shandong, and he told us that the content of the meeting was recorded by the scrum masters in various places. Thus we assigned questions related to the meeting process to two scrum masters to answer during the interview. You can refer to Table 6 below for information on who would answer each question.

The interview questions can be divided into four categories: background, various meetings in Scrum, detailed challenges found in SLR, personal views about tools. All the questions in the interview and motivations were written in Table 5 below. The idea that we put B) before C) was to use B) to disperse their thinking, so that they could come up with more problems they encountered in the meeting process, although not in terms of communication. If the orders of B) and C) were exchanged, the interviewee's thinking might be limited by the four specific issues of communication we found in our SLR, then they were likely to repeat the content mentioned before in the follow-up questions when it came to “meeting”. This was not conducive to our attempts to find other challenges related to communication. Due to the flexibility of semi-structured interviews, the advantage of our design was that if the interviewee mentioned the contents related to the detailed challenges we found in SLR(which should be asked in C)) when discussing the challenges encountered in the meeting process(the B) category), we could flexibly decide to jump directly to ask the corresponding question below or just skipped it when it was the turn of the question below based on what the interviewee answered.

The followings are the questions and their motivations we designed. Capital letter labels represent the classification of questions, number labels represent specific questions, and lowercase letter labels represent the types of answers (ie options).

Table 5.Questions in the interview

Questions in the interview
A) Background: The first three questions are mainly used for warm-up activities, to understand each interviewee's Scrum project experience. (1).How long did you use the Scrum framework before the project? Motivation: We want to know the Scrum project experience level in years everyone has. (2).Have you used Scrum in a distributed environment before this project? Motivation: We want to know if the interviewee has experience in using distributed Scrum development, which is consistent with our theme. We can pay more attention to the answers of experienced people. (3).How did you get in touch with Scrum? Motivation: We want to know the source of the interviewees' knowledge and understanding about Scrum. (4).What challenges do you think the distributed scrum teams face? Motivation: Through this question, we can know whether communication among stakeholders, staff coordination and arrangement, and project control are problems for them. They can also provide us with more problems we may not have. (ex).Can you briefly introduce the work of your two teams in the project? Motivation: To know about the work condition of the two teams and try to get some useful information which may help us to understand what the interviewees said related to the project more easily.

B) Various meetings in Scrum:
5-8 Motivation: The four types of meetings in Scrum are important channels for communication between various stakeholders. Many information exchanges are completed at the meeting. We want to understand their meeting process to find out various problems existing in meetings and how they avoid, respond to, or reduce the impact of these problems. Of course, we can also find some suitable solutions by analyzing their meeting processes and the contents of the answers of various stakeholders which other distributed Scrum teams can learn from. Note: The types of answers and follow-up questions of questions in B) are consistent with question (5.2) except question (5.1), (6.1), (7.1) and (8.1).

(5.1).What is the process of your sprint planning meeting?

(5.2).Have you encountered any problems in the sprint planning meeting?
Types of answer available to interviewees:(we will ask different follow-up questions according to the interviewee's choice)
a).Encountered before:
-Can you describe the detailed issue and what do you think caused this issue? (Get the detailed issue description and compare with our factors to see if we can get new factors)
-And how do you solve this issue later?(get the solution)
b).Still experiencing:
-Can you describe the detailed issue and what do you think caused this issue? (Get the detailed issue description and compare with our factors to see if we can get new factors)
-What do you think can solve or reduce the impact of the issue?(get the solution)
c).Never encountered
-In you opinion, what did you do that avoid this issue?(get the solution)

(6.1).What is the process of your daily stand-up meeting?

(6.2).Have you encountered any problems in daily stand-up meetings?

(7.1).What is the process of your sprint review meeting?

(7.2)Have you encountered any problems in the sprint review meeting?

(8.1)What is the process of your sprint retrospective meeting?

(8.2)Have you encountered any problems in the retrospective meeting?

C) Detailed challenges found in SLR:
Motivation of 9-13: Based on the results of SLR, we extracted four main issues that caused the communication problems of distributed Scrum development. We want to find out from the questions 9-13 whether the development team encountered the same problem in our research case. If so, what did they think that caused the problems? If not, how do they reduce or avoid the impact of these problems?Note: The types of answers and follow-up questions of questions in C) are consistent with question (5.2).

(9).Synchronous communication can be defined as real-time communication between people. Examples include face-to-face communication, video conference and telephone communications. Have you ever encountered a lack or difficulty of synchronous communication in the development process between different teams, such as face-to-face communication?

(10).Have you encountered the issue that team members do not participate or are not actively participating in meeting communication?

(11).Have you encountered the issue that team members do not participate or are not actively participating in ordinary communication?

(12).Have you encountered the issue of inefficiency in communication or non-beneficial communication?

(13).Have you ever encountered the issue of misunderstanding in communication?

D) Personal views about tools:
Motivation: Workers must first sharpen their tools if they want to do their best. We want to know whether the project management tools selected by the team can help their communication.
(14.1).What tools do you use to visualize the progress of each team?
(14.2).How is the experience?

The following table shows the distribution of each question to each interviewee.

Table 6.Questions for each interviewee

Role and its abbreviation	Question numbers to be asked
Scrum master of Shandong(SSM)	(1),(2),(3),(4),(5.1),(5.2),(6.1),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.1),(14.2)
Scrum master of Hong Kong(HSM)	(1),(2),(3),(4),(5.2),(6.2),(7.1),(7.2),(8.1),(8.2),(9),(10),(11),(12),(13),(14.2)
Product owner(PO)	(1),(2),(3),(4),(ex),(5.2),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.2)
Shandong Developer1(SDP1)	(1),(2),(3),(4),(5.2),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.2)
Shandong Developer2(SDP2)	(1),(2),(3),(4),(5.2),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.2)
Shandong Developer3(SDP3)	(1),(2),(3),(4),(5.2),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.2)
Shandong Developer4(SDP4)	(1),(2),(3),(4),(5.2),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.2)
Shandong Developer5(SDP5)	(1),(2),(3),(4),(5.2),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.2)
Shandong Developer6(SDP6)	(1),(2),(3),(4),(5.2),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.2)
Hong Kong Developer1(HDP1)	(1),(2),(3),(4),(5.2),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.2)
Hong Kong Developer2(HDP2)	(1),(2),(3),(4),(5.2),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.2)
Hong Kong	(1),(2),(3),(4),(5.2),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.2)

Developer3(HDP3)	
Hong Kong Developer4(HDP4)	(1),(2),(3),(4),(5.2),(6.2),(7.2),(8.2),(9),(10),(11),(12),(13),(14.2)

3.2.5 Archived data result

a)Project Background：

In 2019, two subsidiaries of the company where we conducted the case study jointly developed an application called DangDuoDuo. The main functions were the sale of financial products, insurance, pawns and financing. The core business was the auction of pawn company bills and corporate finance. In this project, because of the financial background, it was necessary to have sufficient financial knowledge, and it was needed to tap, find potential customers and recommend suitable financial services, so it involved AI and big data algorithms.

b)Organization chart:

There were 8 people in Shandong and 5 people in Hong Kong. Each distribution team had its own scrum master, and the two teams had only one product owner. There were totally 13 people in the distributed scrum teams.The type of team in the case was the second one mentioned in section 2.1.1——Multiple distributed teams and co-located team members as mentioned by Ghani et al.[48] The team members of each team work together in the same location and can achieve face-to-face interaction, while the two teams have geographical distance.

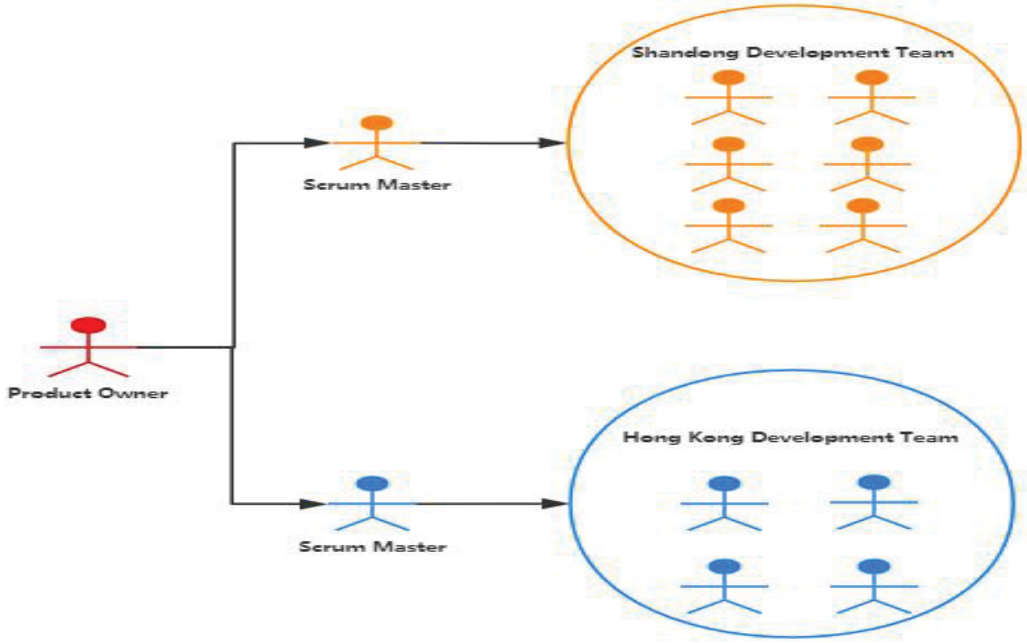

Figure 7.Organization Chart

We asked the Shandong Scrum master to introduce the years at the case company of these employees. As shown in the Table 7 below, we observed that both the scrum masters and the product owner had relatively long working ages. Both teams had a new recruit, and the overall age of the team was relatively young, which was in line with the characteristics of general new startups.

However, in this part, we could only get the working years of the team members in the case company, but not the time of the team members using the scrum. So in our interview, we also asked the participants' actual scrum experience and their knowledge sources of scrum. And we combined these information in the table below.

Table 7. Employee work experience

Role and its abbreviation	Years at case company	Scrum experience(years)	Knowledge source of Scrum	Have distributed scrum experience?
Product owner(PO)	5	7	work	Yes
Scrum master of Shandong(SSM)	3	3	work	Yes
Scrum master of Hong Kong(HSM)	4	4	school	Yes
Shandong Developer1(SDP1)	4	6	work	Yes
Shandong Developer2(SDP2)	3	5	school	Yes
Shandong Developer3(SDP3)	3	3	work	Yes
Shandong Developer4(SDP4)	2	2	work	Yes
Shandong Developer5(SDP5)	2	2	work	Yes
Shandong Developer6(SDP6)	3	3	work	Yes
Hong Kong Developer1(HDP1)	3	3	work	Yes
Hong Kong Developer2(HDP2)	3	6	school	Yes
Hong Kong Developer3(HDP3)	2	2	work	Yes
Hong Kong Developer4(HDP4)	1	1	work	No

c)Meeting time:

Through this meeting schedule, we could know the staff of this company worked six days a week(from Monday to Saturday and Sunday off). The project started on 3rd June,2019 and ended on 20th July,2019. There are three sprints in total and each sprint last 14 working days. In the process of iteration except Sunday, they held stand-up meetings everyday.

Table 8.Meeting schedule

Time	Meeting
2019-06-03	Sprint planning

The first sprint: 14 working days (Saturday is working day in this company, daily scrum)	
2019-06-18	Sprint review
2019-06-18	Sprint retrospective
2019-06-19	Sprint planning
The second sprint: 14 working days(daily scrum)	
2019-07-04	Sprint review
2019-07-04	Sprint retrospective
2019-07-05	Sprint planning
The third sprint: 14 working days(daily scrum)	
2019-07-20	Sprint review
2019-07-20	Sprint retrospective

4 RESULT AND ANALYSIS

4.1 SLR Result

The following contents show the detailed results obtained by the SLR process.

4.1.1 Search Result

After we identified the detailed protocol review (including search keywords and got the search strings), we started to search the articles (the first step of conducting the SLR process) that were related to our research topic in the five search resources.

We decided to use our search strings by searching articles' title, abstract and keywords in "advanced search". Since what we wanted to obtain was the most common issues that the distributed scrum team encountered in the development process and distributed scrum development was also not a very old concept, so we did not limit the publication time of the articles. In addition, all digital databases used for this search supported language standard-based article exclusion, so non-English articles were excluded(I2 and E1, which refer to Inclusion/Exclusion criterias) during the search process. Another advantage of these five digital databases was that they had classified the articles searched so we could easily get all the articles of the types we wanted based on I5 and E5.

According to E2, in collating the literature by using EndNote (which is a tool for organizing and managing literatures), we recorded the publishers of all the literature, and ensured that all of our cited articles were peer-reviewed. Then the following descriptions and Table 9 below shows the detailed screening process and the final screening result after we got the primary articles through the above operations.

The explanation of each step in the Table 9 below:
S1:The number of remaining articles after deleting duplicates with the help of EndNote(E6)(It refers to that we used E6 in this step,so were the I4, E4 and E3, etc. in the below).
S2:The number of remaining articles after reading titles and abstracts to determine if the searched literature is about distributed agile software development(I4 and E4).
S3:The number of remaining articles after reading conclusions and discussions to determine if the searched literature clearly mentions challenges encountered by distributed agile teams(I4 and E4), here we also used I3 and E3 to judge whether the full text could be viewed.
S4:The number of remaining articles after reading the full text to determine whether the searched literature is mainly about scrum(I4 and E4).
S5:The number of remaining articles after merging articles from five databases and deleting the duplicates(E6).
S6:The number of remaining articles after assessing the quality of the selected articles.
S7: The number of articles after doing snowball sampling for the first time(E6).
S8: The number of articles after assessing the quality of the articles from first time snowball sampling.
S9: The number of articles after doing snowball sampling second time(E6).
S10:The number of articles after assessing the quality of the articles from second time snowball sampling.
S11:The number of articles after doing snowball sampling third time(E6).
S12:The number of articles after assessing the quality of the articles from third time snowball sampling.
%: Percentage of articles from different sources.

Table 9.Primary articles selection

Database	Total article found	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	%

IEEE	476	153	66	40	25	17	14	16	16	17	17	17	17	28
Scope	1346	381	109	79	39	13	11	14	13	14	14	14	14	23
spring	226	154	44	28	12	12	8	9	9	10	10	10	10	17
Wiley	560	124	22	7	4	3	3	4	4	4	4	4	4	7
acm	1078	329	107	39	15	3	3	4	4	4	4	4	4	7
others	/	/	/	/	/	/	/	8	7	10	9	11	11	18
Totoal	3686	1141	348	193	95	48	39	55	53	59	58	60	60	100

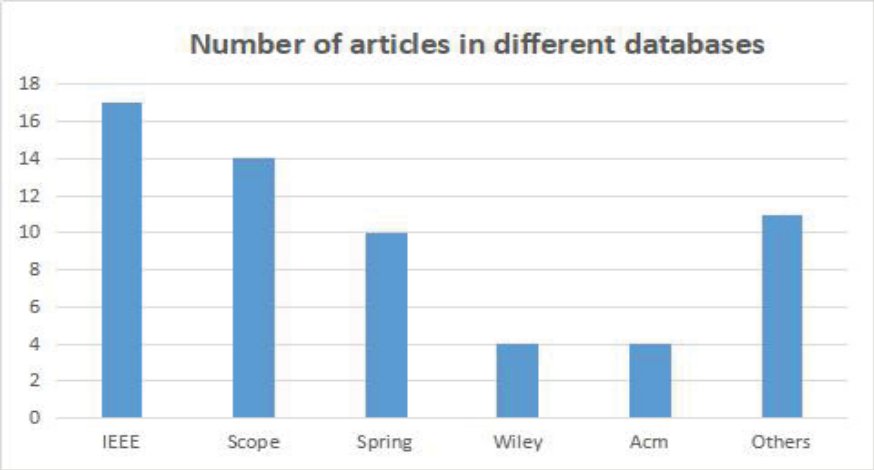

Figure 8.The source of articles

The Figure 8 above shows the source of the articles and the number of articles found in each database. Others refer to Google Scholar, some school websites or academic forums, etc.

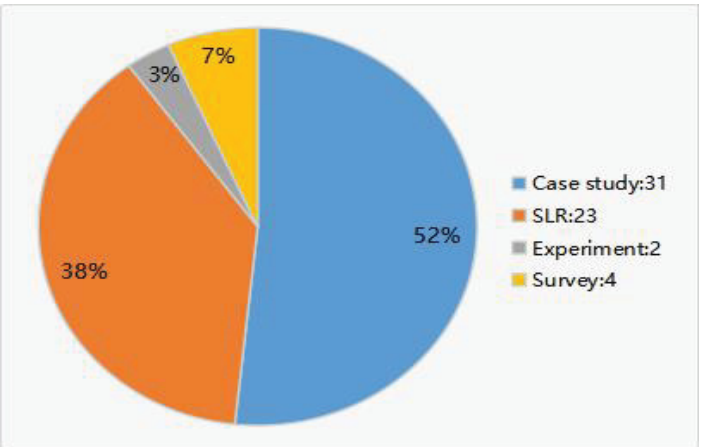

Figure 9.Research type of the article

The Figure 9 above shows the number of research types and the corresponding percentages for each article.

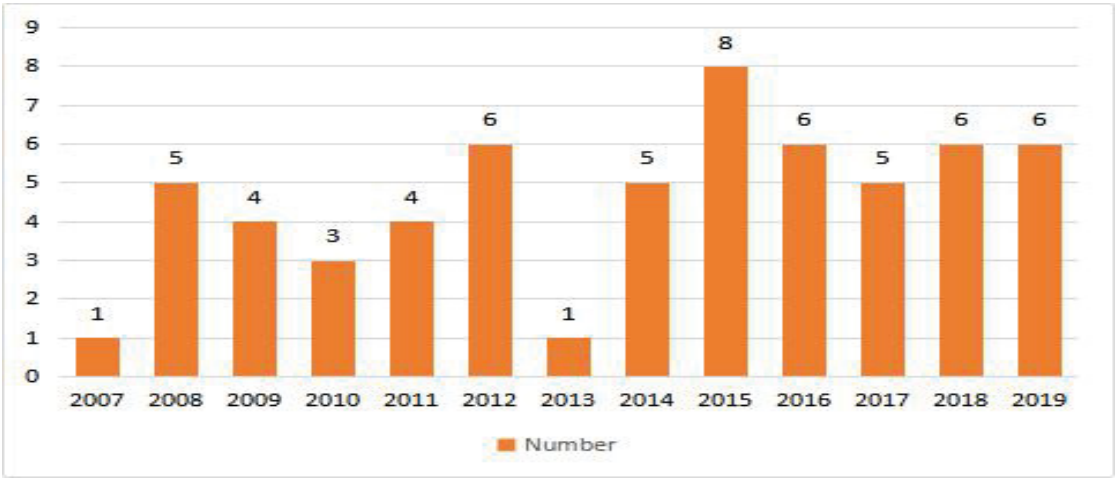

Figure 10.Publication time of all articles

The Figure 10 shows distribution of the years of articles obtained through SLR.

4.1.2 Quality assessment result

The Figure 11 below summarizes the individual studies and their rigor and relevance scores. For a more intuitive view, we used a multidimensional bubble chart to show the amount of distribution of all literature in relevance and rigor. The size of the bubble in the figure was determined by the number. In this figure, according to Munir et al.[95], we defined rigor and relevance categories. Low rigor was defined as 0–1.5 and high rigor was defined as 2.0 and above. Low relevance was defined as a score of 0-2.0, and high relevance was defined as a score of 2.5 or higher.

It could be seen from the figure that the articles marked in red show lower relevance and lower rigor. We would not use these articles to perform data extraction and synthesis. The blue bubbles were the available quality assessment scores and these kinds of articles were used by us to extract data. In the end, we filtered out 12 articles and kept 60 articles. About the detailed information of quality assessment results, you can check the link in the Appendix.

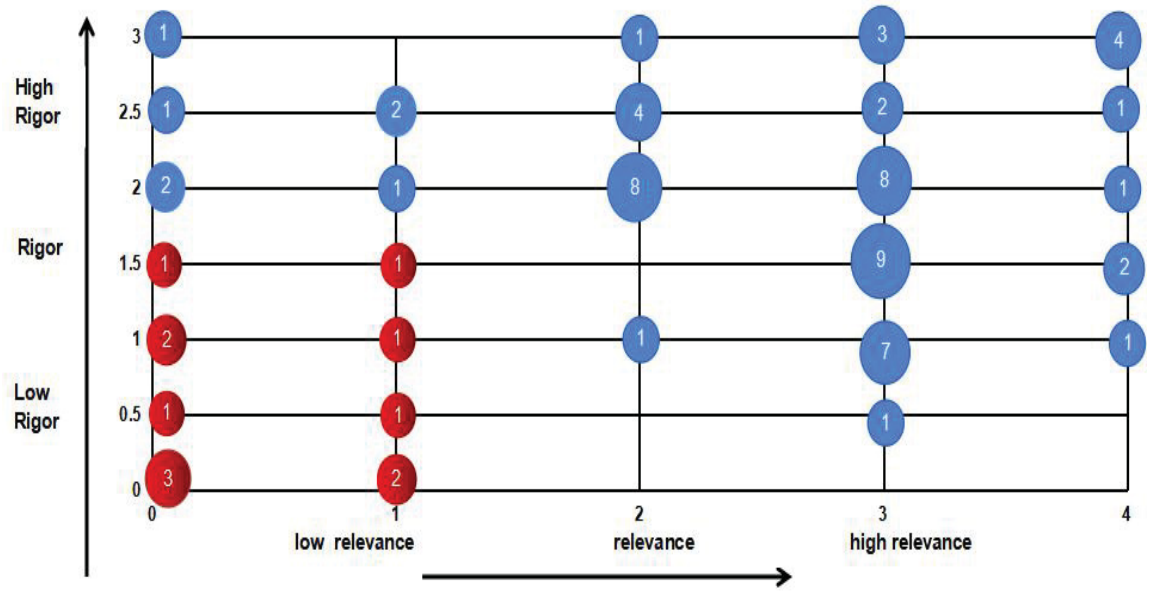

Figure 11.Rigor and relevance bubble chart.

4.1.3 Data extraction and synthesis result

Before extracting the data from the evaluated articles, we had these following discoveries during previous screening and evaluation of the literature:

1.Each article mentioned not just one but many different challenges.

2.Many articles used different descriptions to mention the same type of challenge. We thought that these different descriptions but essentially the same kind of challenge could be summarized under the same name.

3.Most articles listed these issues using a lot of text, while only a few articles categorized the specific issues mentioned according to their common points, which we thought the classified data was more clear and easy for us to understand.

Specifying categories before extracting data would limit the type and scope of the data we extracted, and we did not want to encounter the situation that a challenge we found was out of the categories we defined. So our idea was to first find all the challenges mentioned in the literature we selected, and then classify them according to their common points.

We started data extraction by reading articles that had higher rigor and relevance scores after quality assessment. As mentioned by Shameem and Shrivastava et al.[22][32], we found they relatively listed more challenges encountered by the distributed scrum teams, and we decided to use the names of the challenges mentioned in these two articles as the basis for naming challenges in other articles. That means, if the remaining articles mentioned the similar challenge but using different descriptions, we would use the names in these two articles to name the new found challenges. When we found challenges that were not mentioned in these two articles, we would name them according to our understanding and use these names in the same way mentioned above as the basis for naming the latter similar challenges. During the extraction process, we would appropriately improve the names to make them more consistent with the challenges found.

Then we referred to the same classification criteria mentioned by Agerfalk [69], Rahman [50] and Usman et al.[49] and then defined the three following categories to classify all the challenges we extracted: 1.Communication among stakeholders 2.Staff coordination and arrangement 3.Project control.

The following part shows the details of the categorises we defined and the challenges we extracted.

I.Communication among stakeholders

Description: No matter what type of development team is, communication between people is important. According to Kahya and Seneler [17], communication is an exchange of information between people that can be formal or informal.Based on Carston et al.[100], it is a fundamental and critical process for software development especially for those teams that are in distributed locations. According to Anwer [101] and Schwaber et al.[102], communication can help product owner better understand and deal with the interests of everyone with a stake in the project as well as convey the vision and backlog of the product more correctly. Communication between Scrum master and development team members can ensure that the team is moving in the right direction and following scrum practices, rules and values to gain the business value. Communication between development team members can help them understand each other and share progress so as to gain the success of each iteration and of the project as a whole. These all play positive roles in project development. But the temporal, geographical, cultural distance and other factors caused by the distribution of the team have brought huge challenges to communication among stakeholders. During the communication process, due to the influence of culture or knowledge, team members sometimes can not understand each other's ideas and misunderstandings can easily happen, some members speak little or are even unwilling to speak. Thus many communication processes become ineffective or inefficient. So for many people, they prefer to communicate via email, short messages, etc. rather than face-to-face

communication. But because of time differences, they often can not get the response in a timely manner. Lack of communication can also easily lead to a lack of trust and cohesion among all stakeholders. Then they will become increasingly reluctant to communicate, this is the last thing we want to see during the development process. Table 10 below lists the detailed issues of communication among stakeholders.

Table 10. Issues of communication found in SLR

Issue No.	Issues in articles	Articles that mention the issue	Count
1.1	Lack or difficulty of synchronous communication such as face-to-face communication	[1][2][3][4][5][7][10][12][14][15] [16][17][20][22][23][24][25][27] [29][30][31][32][35][36][39][40] [41][42][46][47][48][49][50][51] [52][54][55][56][57][58][59][60]	42
1.2	Some members do not participate in communication or have low enthusiasm for participation	[3][4][8][10][11][12][13][20][23] [25][32][33][37][38][45][47][49] [50][51][55][57][58][59]	23
1.3	Inefficient or even ineffective communication	[2][3][4][5][6][7][8][9][10][12][13] [14][15][16][19][20][22][23][26] [29][30][32][39][43][44][45][46] [47][48][49][50][51][53][55][56] [57][58][59][60]	39
1.4	Misunderstanding in the process of communication	[2][3][5][6][7][14][17][18][24][25] [32][35][37][38][41][45][49][50] [51][52][53][54][55][56][57][58] [59]	27
Total			131

1.1 Lack or difficulty of synchronous communication such as face-to-face communication

According to Pagani [103], synchronous communication can be defined as real-time communication between people. Examples include face-to-face, video call and phone communication. In synchronous communication, the voice, intonation, body language of the speaker can provide us with a lot of extra information. The biggest feature of synchronous communication is that there is no obvious time delay between questions and answers, and we can directly respond to each other's words and deeds. And an advantage of synchronous communication is that the respondent's answer is more spontaneous, without long thinking according to Opdenakker[104]. As described in the typical example mentioned by Rizvi et al.[23]: time zone differences make it difficult to schedule synchronous communications, especially meetings that require a long period of communication, such as sprint planning. The existence of geographical distance makes the number of face-to-face communication between teams less and more difficult. In addition, if team members do not know each other personally, they are more inclined to communicate asynchronously, which will also lead to a lack of synchronous communication. It is very necessary for all stakeholders to have synchronous communication to improve mutual understanding.

1.2 Some members do not participate in communication or have low enthusiasm for participation

Although many different types of communication are arranged in the organization, there are still some members who are absent or not motivated to participate in communication for some reasons. As mentioned by Gupta and Manikreddy[13], the developers themselves rarely communicate, most of them are committed to achieving their personal goals, and the team does not participate in weekly status meetings, which leads to rework and delays in progress. The Bulgarian team does not share social time with the Swedish team and does not take part in the communications because the geographical distance creates a division among them based on Szabó and Steghöfer [3]. Furthermore, poor communication results will greatly reduce the enthusiasm of members to participate in communication. As mentioned by Paasivaara and Lassenius[8], team members complained that the common meetings didn't help them much, and even saw it more or less as a waste of time to participate. How to make all stakeholders hear everyone's voice and how to increase the enthusiasm of everyone to participate in the communication are both important problems that we need to pay attention to.

1.3 Inefficient or even ineffective communication

The value generated by communication is not always proportional to the duration of the communication. Some communications may not be of much help or even do nothing to the people involved. Due to the language and knowledge gap, members often spend a lot of time explaining during the communication process based on Williams and Stout[105], which will make the communications longer. Lack of suitable communication tools and poor network bandwidth will also cause inefficient communications according to Kajko et al.[14] As mentioned by Therrien[16], because of using the bad tools, both teams could not hear each other during the communication process. It's a common thing that team members don't know what to say during the communication, and they are not interested in what others are doing, sometimes they can't even understand each other's problems, which lead many participants to feel that such communication is useless and even claim that it is a waste of time[9]. When speaker's gestures and facial expressions were not recognised, or attention to subtle information such as the emotion of an individual was not noticed, ineffective communication would be easy to occur according to Dorairaj et al.[55] Besides, staggered time zones also cause inefficient communications. Team members send mails but can not get a response in a timely manner as mentioned by Beecham et al.[36]

1.4 Misunderstanding in the process of communication

Due to the knowledge gaps and stakeholders do not understand each other's culture, misunderstandings often occur during the communication process. Misunderstanding of requirements and project knowledge will cause lots of rework, and misunderstanding of politics and religion will cause some unnecessary conflicts. According to Paasivaara et al.[25], misunderstandings are very common. Developers may have a totally different picture in his head about some functionality than a product owner has. Sometimes this misunderstanding is not revealed until the sprint demo, which will cause many rework and waste a lot of time. An interesting example of cultural misunderstanding is: In India, the answer of "yes" means "yes, I heard you", while in America, "yes" means "yes, it's done" as mentioned by Dorairaj et al.[55]. In addition, differences in political, religious or moral values can also easily cause misunderstandings among colleagues according to Kahya[17]. What is different from "inefficient or even ineffective communication" is that "misunderstanding in the process of communication" refers to you do not waste a lot of time in the communication but the information you get is not what the other person really wants to express, while "inefficient or even ineffective communication" focus on expressing you have wasted a lot of time in explaining many things that are not necessary to explain, but the result of the communication is that both sides get right information.

II.Staff coordination and arrangement

Description:Coordination refers to bringing together the knowledge, experience, and skills of team members to achieve the collective goals of the project based on Asproni[106]. In distributed agile development, team members usually come from different regions, have different cultural and educational backgrounds, and often work in different time zones, so coordination will be costly. The

most difficult thing in coordination is the assignment of tasks. Tasks assigned to geographically distributed team members requires strong coordination among them, as team members are interdependent according to Kaur and Sharma[58]. This leads to difficulty in task assignment during development. At the same time, distributed teams are often composed of people from different countries and cultural backgrounds, so they all have different working habits, which reduces the willingness of team members to cooperate. It is not conducive to the development of team culture, can reduce the team's cohesion, and make it difficult to arrange team members for activities. Due to different levels of knowledge and different educational backgrounds, this also makes team members often misunderstand or perform improperly. The following table gives the detailed issues regarding coordination.

Table 11. Issues of coordination found in SLR

Issue No.	Issues in articles	Articles that mention the issue	Count
2.1	Different work practice standards or work habits	[3][6][11][13][14][19][29][31][32][33] [36][38][48][49][51][53][55][56][58]	19
2.2	Difficult or improper assignment of team tasks and workload	[3][8][22][23][27][30][32][35][42][49] [50][51][52][53][56][59]	16
2.3	Team members are less likely to see themselves as part of the team	[3][8][9][12][17][30][31][35][45][47] [50][51][53][55][56][58][59]	17
2.4	Team members do not understand the assigned work and responsibilities well or perform them poorly	[4][10][13][14][15][17][18][21][22][23][24][31][35][40][41][45][51][53][56]	19
2.5	Difficulty or improper arrangement of team activities	[3][4][6][7][8][9][11][12][14][16][17] [19][20][22][23][24][26][34][35][36] [40][41][48][50][51][55]	26
2.6	Difficult to change team member structure	[11][13][15][24][31][40][56][59]	8
Total			105

2.1 Different work practice standards or work habits

Because distributed teams often come from different countries and cultural backgrounds. When project work is assigned, there exists cultural issues such as language barriers, work cultural differences, cultural prejudice, etc. These factors may have a devastating effect on the implementation of the project based on Shrivastava and Date[51].

2.2 Difficult or improper assignment of team tasks and workload

It is difficult to assign work tasks and perform performance in distributed scrum teams. Due to the influence of geographical distance and the different levels of knowledge of team members, it is possible that managers may not know enough about foreign district workers. It was mentioned by Sievi et al.[35] that the work arrangements in the project should be clearly defined. The lack of understanding of each other's skills and other backgrounds caused problems in the first half of the project. So more consideration should be taken when assigning the tasks.

2.3 Team members are less likely to see themselves as part of the team

In distributed development, workers in different regions are unlikely to see themselves as part of the same team due to sociocultural distance. It is difficult for team members to trust the good will of remote colleagues and may lack team cohesion according to Dullemond et al.[47] These things make it more difficult to work closely with developers in different geographical locations.

2.4 Team members do not understand the assigned work and responsibilities well or perform them poorly

Trust and cohesion are not strong due to differences in knowledge levels and geographical and cultural distance. In distributed scrum development teams, untrained members are less concerned with standards and organizational purposes. They always consider their unique goals as mentioned by Zada and Shahzad[52]. This can lead to them not understanding their job responsibilities or performing poorly in their work and failing to complete the task.

2.5 Difficulty or improper arrangement of team activities

In distributed agile development, because they are not working in the same area, the two teams can often only communicate asynchronously, which is a great challenge for event, time and staff scheduling. For example, it is not that easy to properly arrange meetings between teams and arrange mutual visits between teams. On the other hand, this places high demands on professional literacy between different teams. If developers lack the extended knowledge and awareness of agile methods, they are more likely to fail to implement agile successfully, which lead to project failures and failures according to Moe et al.[41]

2.6 Difficult to change team member structure

The capabilities and knowledge of leaders and practitioners are important factors in extending agile practices in a distributed environment. Based on Shameem et al. [22], we know that when an agile practitioner or manager who is highly committed to implementing agile practices in DSD is replaced by a person with a lower commitment, all previous activities will be lost. When someone leaves or joins a team, it takes a lot of effort to maintain a high level of trust within the team according to Moe et al. [41] So when the culture of a team and the efficient coordination of a team are established, changing the team structure becomes difficult.

III.Project control

Description: Finally, according to our literature, time, geography, and sociocultural distance may affect the GSD control process. In agile projects, the emphasis on up-front planning and strict control has been reduced, and more reliance has been placed on informal collaboration, coordination, and learning as mentioned by Dybå and Dingsøyr [107],However, in distributed development, it is difficult to achieve good project control. Due to the influence of geographical distance, the integration of the project poses a great challenge. And because of cultural differences, employees may not be comfortable with their work and dissatisfied with the office environment during the project. Table 12 below lists the issues related to project control in detail.

Table 12. Issues of project control found in SLR

Issue No.	Issues in articles	Articles that mention the issue	Count
3.1	Difficulty in project integration	[6][9][11][12][15][17][20][29][31][35][36][48][52][53][56][57][58][59]	18
3.2	There is a lot of rework in the project	[11][13][40][41][57]	5
3.3	Team members are not satisfied with the work and office environment in the process of the project	[6][12][31][56][59]	5
3.4	Insufficient transparency or visualization of the project	[3][4][6][12][16][20][23][34][41][55]	10
3.5	Unreasonable determination of project requirements and priorities	[8][18][20][24][27][28][30][49][59]	9
3.6	There is a certain gap between the achievement,cost of the project development and the expectation	[13][16][18][20][22][23][26][36][41][46][47][53][59]	13
3.7	Improper management and sharing of project documents	[1][4][6][10][11][13][15][17][18][21][22][23][24][26][27][32][34][51][55]	19
Total			79

3.1 Difficulty in project integration:

In distributed software development, a project will be divided into many small modules and assigned to various teams. Project integration can be considered as all the business units of one organization use a consistent approach to project management and share information about project requirements and objectives according to Kirsilä et al.[109] Due to the poor project experience, the gap of knowledge understanding and the lack of project integration awareness, etc, the teams are likely to face many difficulties in the project integration phase. As mentioned by Beecham et al.[36], the lack of clear, company wide deployment and integration processes makes project integration difficult for

inexperienced teams. Lack of continuous integration is a major challenge for continuous deployment, as the project integration phase can take several weeks before each release, which reduces team efficiency and slows down the overall progress of the project based on Paasivaara et al.[31]. Project integration is an important role in any project's success. It can ensure that all of the elements that are necessary to complete a successful project come together according to Yazdanifard et al.[108]

3.2 There is a lot of rework in the project

In distributed agile development, there will always be a lot of rework problems. Due to different levels of knowledge or inadequate understanding of technology, developers have made mistakes in development. Or it is difficult to communicate information to developers in a timely manner due to the time distance. So it caused a lot of rework. As the example mentioned by Cristal et al. [11]: The lack of management maturity in agile practices affects the development life cycle. To reduce the scope of the sprint, management decided to discontinue functional testing. This immature decision caused the team to rework.

3.3 Team members are not satisfied with the work and office environment in the process of the project

In our literature search we found that many people were complaining about their office work environment. This was due to higher requirements for communication facilities and hardware in the distributed scrum team. In a survey question mentioned by Cho[56], one participant said "Sometimes VPNs are too slow and can cause frustration." Although through network video, voice calls and various collaboration software can minimize the impact of face-to-face. But it would be much easier to work together in the office according to Cho[56]. So when the geographical and time distance cannot be changed, the high demands of the team members on the office environment will not stop.

3.4 Insufficient transparency or visualization of the project

In distributed development teams, team integration and visualization of projects are difficult. It is difficult for different teams to communicate the progress of the project, and a visual tool is needed to monitor the completion of the project. Through Paasivaara et al.[20], the development team used the backlog management tool Agilefant, and each team could track all the stories, progress, and contributions of each member to its development. However, since the project team comes from different regions, it is often difficult to unify which management tool is used, so it is difficult to ensure the transparency and visualization of the project.

3.5 Unreasonable determination of project requirements and priorities

Analyzing and determining requirements is a key factor for successful software projects according to Boehm and In[110], which can help to reduce the waste of time and optimize software products as mentioned by Liu et al.[111] Then through solving high-priority requirements first and then low-priority requirements,the costs and duration of the project can be significantly reduced based on Hofmann and Lehner[112]. So it is important for us to identify the project requirements and priorities. As mentioned by Paasivaara et al.[20], there can be too many requests and lack priority in the whole project and the product owner may require a lot of user stories in an iteration. That means the teams may not deliver all requirements on time and the storys need to be postponed. So the teams need to select requirements based on their capabilities and then prioritize them appropriately.

3.6 There is a certain gap between the achievement,cost of the project development and the expectation

The team will set an expectation for the results and costs of each iteration in the development process of the whole project, but for some reasons, such as the difference in knowledge understanding, there will be a certain gap between the results and costs of the iteration and the expectation. As mentioned in [23], about project timelines and milestones, some people would like to use project schedules as guidance rather than commitment due to their cultures. So it leads to the gap between development results and expectations.

3.7 Improper management and sharing of project documents

Due to geographical distance and time distance, it is basically impossible to achieve frequent face-to-face meetings or voice calls in a distributed scrum team. The lack of close collaboration between distributed teams results in increased documentation. Teams in two different locations must rely on documentation to communicate according to Kajko et al.[14] This is the opposite of the agile value of less documentation. On the other hand, because the level of knowledge of team members is not used, knowledge sharing can often only be done through documents during the development of distributed projects. Therefore, higher requirements are imposed on file management and sharing in agile projects.

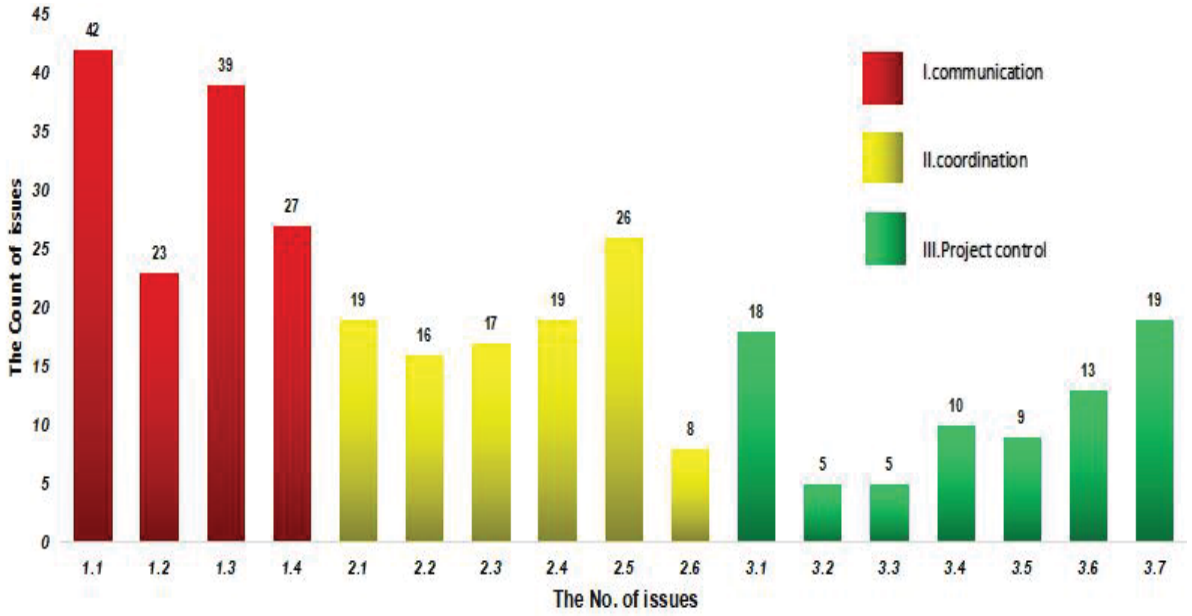

Figure 12. The detailed issues of three different kinds of challenges

The Figure 12 above shows all the issues we found in SLR and the corresponding number.

Then we reviewed the literatures to find the factors that cause these most common issues, this partly answers the RQ2. The results are shown in the table 13 below.

Table 13: Factors found

Factor No.	Factor name	Articles mention this factor	Count
F1	Team members have insufficient knowledge or different skill levels	[2][6][10][15][20][22][24][33][35][41][43][44][45][52][54][58][59][60]	18
F2	Difficult project knowledge sharing and synchronization	[2][23][29][35][43][54][59]	7
F3	Differences in team members' understanding of knowledge	[2][8][9][14][18][24][25][35][37][41][43][45][52][54][56][59]	16
F4	Travel costs are too high	[1][3][4][10][15][16][22][29]	8
F5	Lack of suitable communication tools	[4][5][6][7][12][14][15][16][20][22][23][25][29][31][32][35][37][39][46][47][48][51][52][55][57][58][59][60]	28

F6	Are not familiar with each other or have differences	[3][7][8][9][10][11][12][13][14][16][17][20][23][25][26][29][48][49][50][51][55][56][57][58]	24
F7	Unavailability of stakeholders or failed to respond in a timely manner	[3][13][14][15][18][20][22][24][27][32][44][47][50][51][55][56][57][58]	18
F8	Lack of trust or cohesion	[3][8][10][12][13][14][16][17][20][22][26][30][31][32][35][37][38][45][47][49][50][53][55][56][58][59]	26
F9	Team members do not speak or speak very little during the communication process	[5][9][10][14][20][25][50][55][56][57]	10

1.Team members have insufficient knowledge or different skill levels: Team members are not rich in experience or knowledge in a certain field, and there is a certain gap in their skill level due to different educational backgrounds,etc. In other words, for the same problem, different people show different abilities. Some are good at it, while others are not. This tends to refer to how much of your previous work experience and your mastery of professional knowledge in the software area.

2.Difficult project knowledge sharing and synchronization: It is important but also difficult to synchronize as well as share project knowledge and scrum experience,etc in the whole development process, which can ensure that the entire team avoids detours, uses the correct practices and moves in the right direction. This tends to mean that it is difficult for all members of the team to have a consistent understanding of the various knowledge required in the current project and the lessons learned during the project.

3.Differences in team members' understanding of knowledge: There may be multiple solutions to a math problem. During the development process, when encountering the same technical difficulties, different team members may propose different solutions. At this time, they should have discussions, which can help team members understand each other's ideas and decide how to solve them. And it also reduces the possibility of misunderstanding or disagreement later.

4.Travel costs are too high: In distributed software development, teams are in different locations. They are not able to communicate face to face at all times.If they want to travel to another team’s site for live communication, then they may need to pay a lot of money, time and energy.

5.Lack of suitable communication tools: In order to exchange information with each other, teams sometimes choose to make voice or video calls if they have the equipments. Unclear picture, unclear voice and poor network bandwidth will lead to poor communication between teams. So having good communication tools can greatly increase the efficiency and quality of communication between teams.

6.Are not familiar with each other or have differences: Because the number of collaborations between or within teams is small (probably the first time), team members may not know much about each other's personality, temper, hobby, etc. Then religious, language(accent and pronunciation) or political differences can also lead to differences among team members and even unnecessary conflicts. In addition, in the process of developing projects, team members who do not understand each other ’s modules or work content will also lead to communication obstacles.

7.Unavailability of stakeholders or failed to respond in a timely manner: In distributed agile project development, it is difficult for team members to implement the principle of frequent face-to-face communication in agile. Because they are not in one location, they can only choose video

conferences or voice calls. However, due to the influence of time and other factors, it is difficult to get timely responses from stakeholders.

8.Lack of trust or cohesion: Team members from different cultural backgrounds often face difficulties in understanding each other's culture. Lack of cultural understanding can affect team building trust and cohesion. When the team lack trust, it often make the team members feel fragile in the team and make the team members feel less at work. On the other hand, because different teams have difficulty communicating face-to-face in different regions, they can only communicate through video or telephone, which exacerbated the distrust of both parties. This is very detrimental to the communication of the project, and may cause the project results to be different than expected or even fail.

9.Team members do not speak or speak very little during the communication process: In the communication process of the team, because the distributed development team is difficult to share knowledge, and because the team members come from different cultures, they may also have language barriers.This may result in only one of the two teams saying a lot and the other saying less. Silence makes it difficult to understand what the other person wants to express.

From the analysis above, it was evident that " communication among stakeholders " was the most common challenge faced by DSD practitioners. More than 90% of the articles found in the literature cited communication as the most common challenge. However, other challenges like " staff coordination and arrangement" and "project control" were also cited to be big issues. And nine factors were found by us that could cause this most common challenge through SLR(see Table 13 for description).

4.2 Case study result

We interviewed the 13 interviewees by phone in the order listed above. During the interview, we basically asked the interviewees one by one according to the order of questions we designed, and adjusted the order of questions if necessary as what we have discussed in Section 4.1.4. Through interviewing the interviewees' background information, we learned that everyone had experience in distributed Scrum project development except for one member in Hong Kong.

Through analyzing the interview results, we mainly found the following two problems:

1.Not all questions can be answered by the interviewees, especially regarding the solutions.

2.Some of the contents that the interviewees answered have nothing to do with the topic of our thesis or mentioned something that was not about communication.

Since our goal of the interview was to find out challenges, factors and solutions related to communication, we paid more attention to the content related to communication when we extracted useful information from interview results.

The answers of questions 1 to 3 were listed in Table 7 above. In question 4, we found that the challenge mentioned most frequently by the interviewees was also communication. And through question 14, we found that the interviewees were basically satisfied with the project management tools(TAPD) they used.

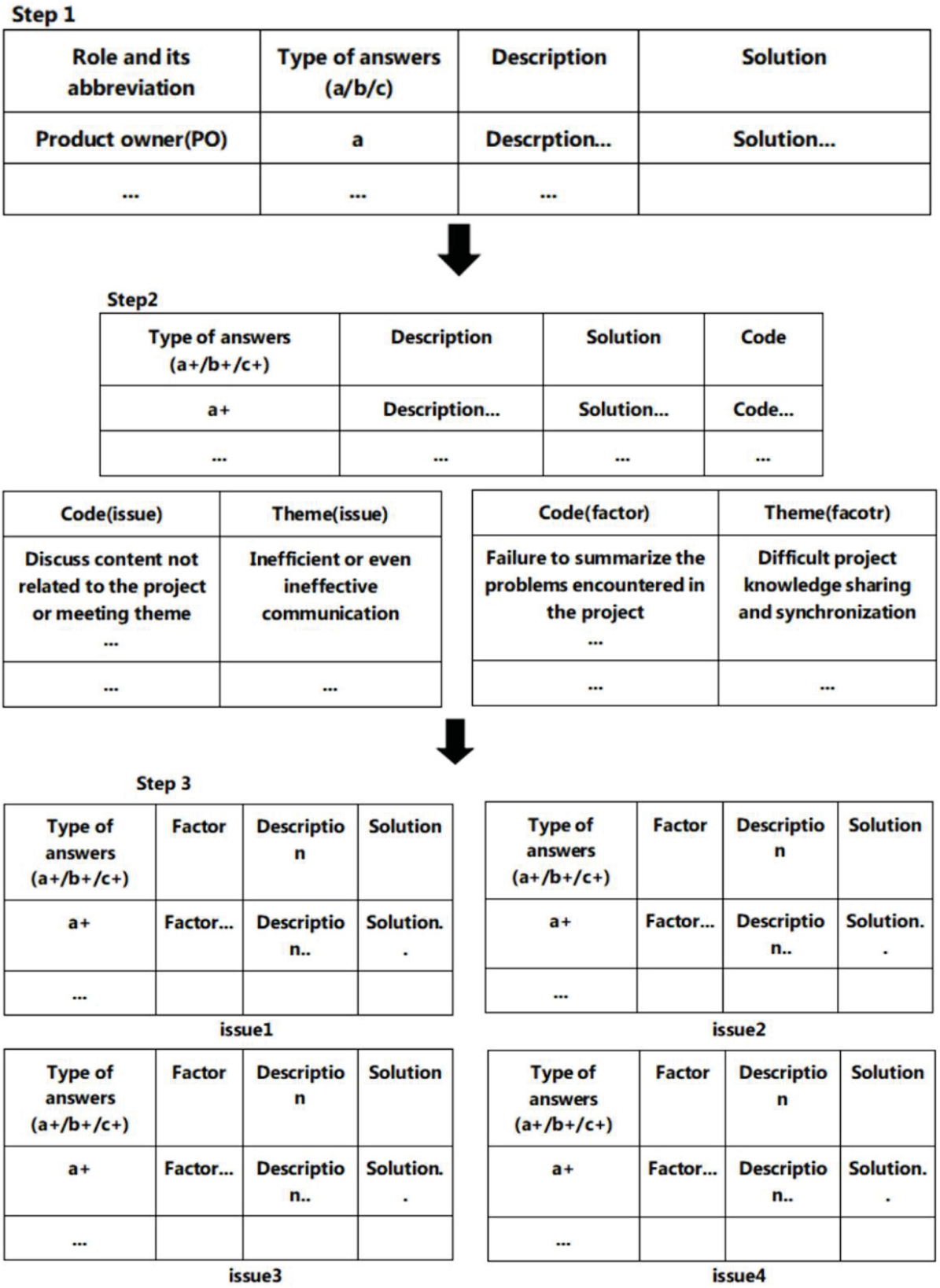

Figure 13.Interview results integration process

When we analyzed questions 5-13,which were our main source of information related to our research topic, we mainly used thematic analysis method to analyze our interview result based on Braun and Clarke's[116] guidelines.

As shown in the Figure 13 above, in Step 1, after we obtained the original answer from the interviewee, we first conducted a preliminary integration and classification of all answers based on the answer type selected by the interviewee and whether a solution was provided. For the specific classification criteria, you can see Table 15 below. During this period, we chose to keep the answers that provided solutions (a + / b + / c +) and discarded those that did not provide solutions (b- / c-).

Then in Step 2, since our initial purpose was to find the types of issues encountered by the interviewees, the factors and solutions, we extracted all related phrases and sentences during the process of traversing and analyzing the answers of all interviewees. These contents helped us understand what the interviewee wanted to express, and also allowed us to quickly classify their answers. Based on our understanding, we first summarized the descriptions of issue types and factors with similar meanings, because sometimes different interviewees would say similar things when answering questions, but used different descriptions. Therefore, here we used these extracted phrases or sentences that can be used as classification criteria for coding. The only difference between questions 5-8 and 9-13 is that in questions 5-8, we obtained some additional codes for detailed issue descriptions. We compared these codes with the four detailed issues related to communication found in the SLR to determine whether the issue can be attributed to one of the SLRs or can be identified as a new problem. But we did not have to do this in questions 9-13, because the issue type had been clarified when designing the question (the four detailed communication issues identified in SLR).

After obtaining the codes and analyzing it with the data in the SLR, we identified 4 themes of communication issues (no new communication issues were found during interviews) and 16 themes of the factors (9 were from SLR, 7 were newly discovered and summarized). Then according to these, we had further classified and sorted all the data, so as to obtain the table shown in Step 3.

Finally, because the solutions needed to be analyzed in conjunction with the detailed issues descriptions, here we didn't use thematic analysis to integrate solutions. We carefully read the description of the issue in each table to determine whether there were similar descriptions in the same table. If yes, then we read these issue descriptions to see if the corresponding solutions also used similar descriptions. If both of the above conditions were met, we chose to merge these same interviewees' answers and used simple descriptions to summarize the issues interviewees mentioned and the corresponding solutions based on our own understanding.
For the original answers of the interview, the detailed information of coding and theme, you can view them through the links in the Appendix.

The following part is the detailed information of the interview results after our analysis, extraction and integration.

The two scrum masters were asked questions about the meeting process.The processes are described as follows:

Table 14.Description of Scrum meetings

Meeting type	Answers of the meeting process
Sprint Planning meeting	SSM: In the planning meeting, product owner will discuss the requirements with all the team members. After ensuring that everyone's understand is consistent we will work out a Sprint backlog. Then after each item in the backlog is broken down into multiple tasks, team members will claim the tasks and complete the estimation of work hours.
Daily standing meeting	SSM: At the specified time, both team members will arrive at the meeting room and speak in the specified order facing the display screen. Everyone just needs to say what they did yesterday, what they will do today and what obstacles they

	encountered. In addition, everyone should not be interrupted when they are speaking, and I will record the troubles encountered by each member for discussion after the meeting.
Sprint Review meeting	HSM: We will conduct a review meeting at the end of each iteration to demonstrate our team 's results to our project leader. Check if we have completed our goal and propose a new product Backlog based on feedback.
Sprint Retrospective meeting	HSM: We will have a retrospective meeting after the end of each iteration cycle. In this meeting, we mainly talk about the good and bad things done in this iteration so that we can continue to get better in the next iteration.

In the interview questions we designed, we provide three different answers for interviewees to select and conduct in-depth inquiries based on their choices.
In order to facilitate our conducting statistics and making charts, we designed specific abbreviations for each type of answer as shown in the table below.

Table15.Abbreviations and descriptions of different types of answers

Three Original answers we provide in interview question	Abbreviation	Descriptions of answers(which original answer did interviewees choose and what kinds of solutions did they provide)	Name of the kinds of solutions(if any)
a).Encountered before	a+	The interviewees selected the answer a) and provided the solution they used.	practical solution
b).Still experiencing	b+	The interviewees selected the answer b) and provided the solutions they envisioned but are not put into effect.	envisaged solution
	b-	The interviewees selected the answer b) but did not provide the solution.	/
c).Never encountered	c+	The interviewees selected the answer c) and provided the precaution they used.	practical precaution
	c-	The interviewees selected the answer c) but did not provide the precaution.	/

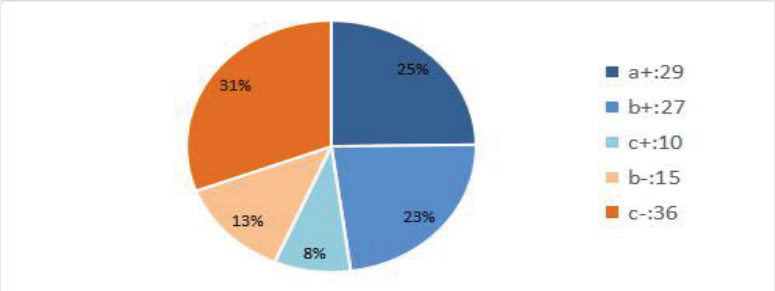

Figure 14.Proportion of different types of answers

The total number and percentage of each type of answer were shown in the Figure 14 above. We extracted totally 96 answers, which we thought were related to communication.

The number of interviewees who thought that they had not encountered relevant challenges and had not given solutionis the most, which accounted for 31%. And we could see a total of 56% of the

interviewees gave practical solutions, envisaged solutions or practical precautions. Among them, a+ is the most which means that the interviewees have used the solutions to solve the challenges they encountered. The remaining interviewees answered that they encountered the challenges but did not give the solution they envisioned.

In addition to the information shown in the above figure, we have also classified these answers according to the four questions we found in the SLR since we do not find other challenges related to communication during the interview. The details are shown in the Figure 15 and table below.

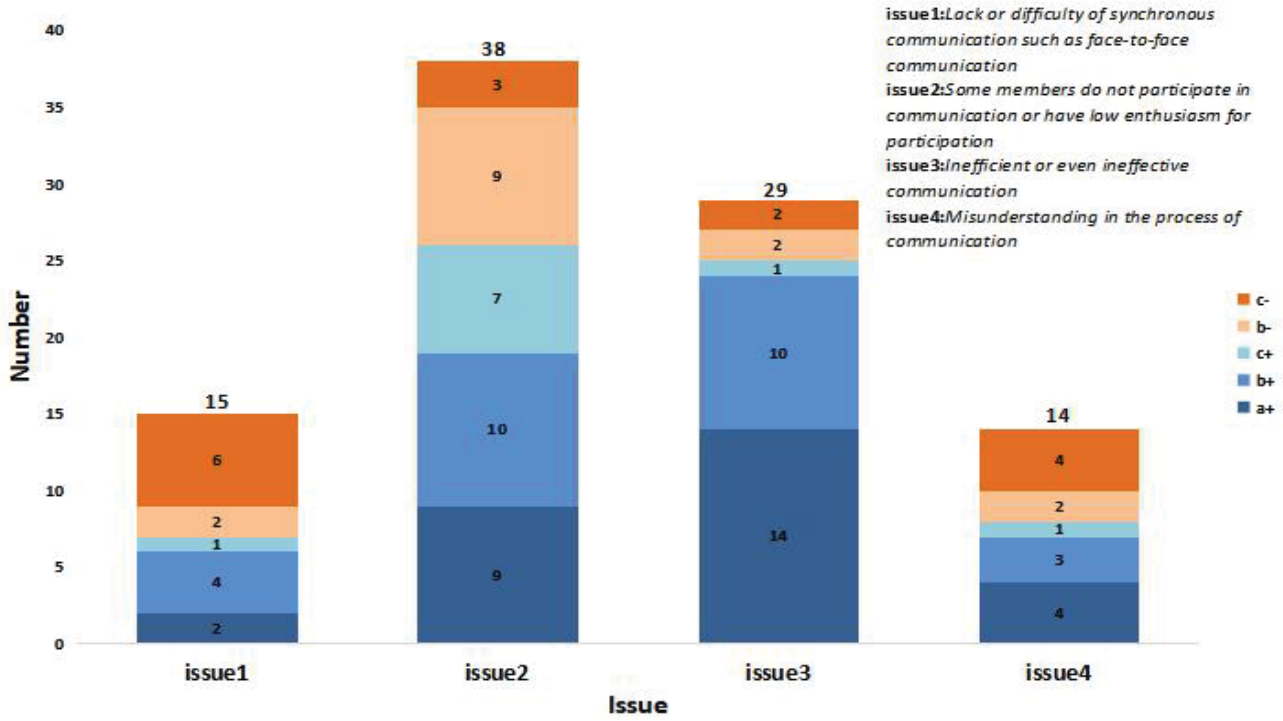

Figure 15.Details of the four challenges related to communication

According to the bar chart above, we observed that issue2 was the most challenge they encountered or mentioned, issue3 was the second. The interviewees gave the most solutions which they actually used on how to solve issue3. They had the same number of envisaged solutions to problems 2 and 3, but were not be applied in practice. And through calculation, we could know for these four issues, more than half of the answers from interviewees provided the practical solutions, envisaged solutions or practical precautions except the answers of issue1 which was only 47%.

The specific number and corresponding percentage of each answer of the four detailed issues related to communication found in SLR are shown in the four pie charts below.

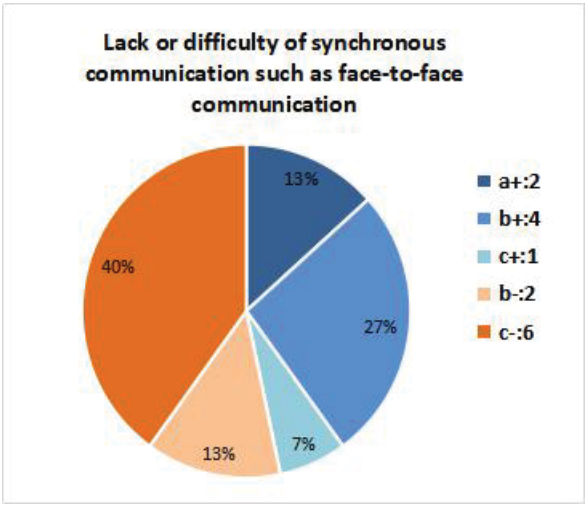

Figure 16.The answers to issue1.1

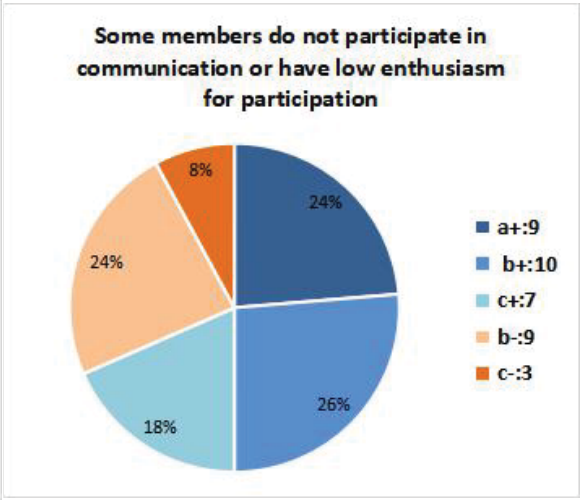

Figure 17. The answers to issue1.2

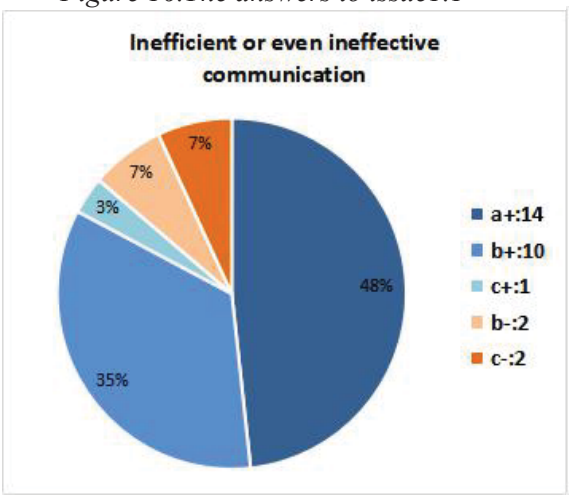

Figure 18.The answers to issue1.3

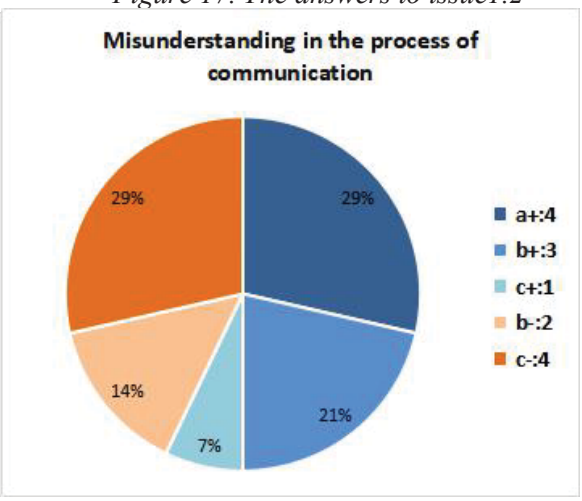

Figure 19.The answers to issue1.4

It is obvious that in Figure 15, the largest proportion of answers indicate that they have not encountered issue 1 and did not give a solution. In Figure 16., half of the answers indicated that I had encountered issue 2, and 68% of all answers gave solutions.In Figure 17, nearly 50% of the answers said that they have encountered and resolved issue 3, and 86% of people gave a solution to issue3.In Figure 18, 50% of the answers indicated that they had encountered issue 4, and 29% of the answers indicated that they did not find this issue 4 and provided no solution.

In the two tables below, we calculated the number of all factors that caused the above four problems. The nine factors shown in the first table are all factors obtained in our SLR. From F10 to F16 in the Table 17 are the new factors we obtained from the interviewees by thematic analysis, and we described them.

Table 16.Factors found in SLR

No.	Name	count
F1	Team members have insufficient knowledge or different skill levels	13
F2	Difficult project knowledge sharing and synchronization	13
F3	Differences in team members' understanding of knowledge	4
F4	Travel costs are too high	2

F5	Lack of suitable communication tools	3
F6	Are not familiar with each other or have differences	12
F7	Unavailability of stakeholders or failed to respond in a timely manner	1
F8	Lack of trust or cohesion	6
F9	Team members do not speak or speak very little during the communication process	8

The detailed description of these new found factors are shown in the table below

Table17.Abbreviations and descriptions of different types of answers

No.	Name	Description	count
F10	Off topic in meeting	Discuss at the meeting without permission, argue about something or talk about something unrelated to the project in the meeting.	12
F11	policy	There are some formalities to go through if team members from Shandong want to visit the team in Hong Kong.	1
F12	Improper meeting process	Improper regulations on equipment commissioning, personnel speaking order, station position and speaking content of the meeting.	9
F13	Team members fail to perform their personal duties	Team members do not perform their duties as employees: such as not paying attention to the lectures, not speaking in the specified order, interrupting others, being late for the meeting, etc	4
F14	Team members lack a sense of participation	Due to the limitation of geographical location, in several important meetings, one team is often the leader, while the other team participating can only participate in the meeting through video conference. This caused many participants to say that their work was neglected and lacked a sense of participation in the entire project.	5
F15	Different work practice standards or work habits	During the development process, two teams that have not been in contact may have their own work styles, such as different completion standards and work habits. This would have an impact on the handover of later work. Especially when the project is relatively large.	2
F16	Tense working atmosphere	In a distributed scrum team, two working teams in different positions are both cooperative and competitive. Both teams can see the progress of the other team in their work software, which virtually increases the sense of competition between the two teams. Too much tense work atmosphere will have a bad influence on the communication between the teams.	2

The following tables are our statistical results about the interview.

Table18.issue1.1：Lack or difficulty of synchronous communication such as face-to-face communication

Type	factor	description	practical solutions/ envisaged solutions/practical precautions
a+	policy	Because of policy restrictions, it was difficult for the mainland team to go to Hong Kong to have face-to-face communication.	Arrange the Hong Kong team to visit Shandong.
a+	Lack of suitable communication tools	The two teams used different social software in the previous life and work.	Unified use of a communication software at work——Wechat.
b+	Unavailability of stakeholders or failed to respond in a timely manner	The customer was busy and has little time to communicate with the team and view the project progress.	More proactively contact the customer and exchange work progress with him.
b+	Travel costs are too high	Having face-to-face communication can cost a lot(time, money and energy).	If necessary, the person in charge of Shandong can help arrange the Hong Kong team to visit Shandong.
b+	Are not familiar with each other or have differences	Accent problems made it more difficult to understand each other on the phone or video.	Sometimes asynchronous communication may be better, such as sending WeChat or emails.
b+	Lack of suitable communication tools	Network fluctuations lead to poor video conference communication quality.	Request the other party to repeat appropriately or view the meeting record after the meeting.
C+	/	/	Buy related equipment and regularly maintain it.

Table19.issue1.2: Some members do not participate in communication or have low enthusiasm for participation

Type	factor	description	practical solutions/ envisaged solutions/practical precautions
a+	Team members fail to perform their personal duties	Sometimes someone was late and could affect the end and start time of the meeting.	Some improvement measures were made for latecomers, such as buying drinks for everyone.
a+	Different work practice standards or work habits	The code style was not carefully determined at the initial stage of the project, and the codes of the two teams were not readable by each other.	Unify the code specification and save it in the teamwork software.

a+	Lack of trust or cohesion	Team members can fall into the state of "fighting meetings" and in the most of retrospective meeting time, they only discuss the bad things that they have done , which affects the morale of the team.	Not only communicate the problems that occurred, but also exchange the good measures made in the previous iteration to improve morale.
a+	Team members fail to perform their personal duties Lack of trust or cohesion	Speaking very little at the meeting and not willing to remember what everyone said deliberately, this can lead to distractions.	Have a preparation before meeting and check the record of the meeting if necessary.
a+	Team members have insufficient knowledge or different skill levels Team members do not speak or speak very little during the communication	Fear of being looked down upon by others for lack of skill or knowledge which is caused by self-respect.	More informal meetings as a supplement to exchanges.
a+	Are not familiar with each other or have differences Lack of trust or cohesion	The two teams collaborated for the first time and there are differences in culture,ages and customs,etc. among team members, which makes it difficult to integrate with each other.	Organize and participate more in group activities like dinner and coffee talks which can help improve mutual understanding.
a+	Team members do not speak or speak very little during the communication Lack of trust or cohesion	The enthusiasm for the daily meeting faded.	Organize more small group discussions to enliven the working atmosphere.
b+	Lack of participation	The review meeting was held in Shandong Province and the members of the Hong Kong team did not attend the meeting on the scene(via video tools), which made the Hong Kong team feel that they lacked the sense of participation.	The Hong Kong team appointed representatives to attend the review conference in Shandong.
b+	Team members do not speak or speak very little during the communication	Sometimes the daily meeting time was very short and the content was few. There were little things that needed to be recorded.	Understand the status of the project before the meeting to better control the rhythm of the meeting.
b+	Lack of participation	Not familiar with the business module and feel a lack of	Consult colleagues or discuss in informal meetings.

	Team members have insufficient knowledge or different skill levels	participation.	
b+	Improper meeting process	Sudden silence during the video conference.	Clarified the unified meeting process, arranged the speaking order and standing position.
b+	Team members have insufficient knowledge or different skill levels Difficult project knowledge sharing and synchronization	Different levels of knowledge lead to a lot of communication without progress, which affects team morale.	Induction training before work.
b+	Tense working atmosphere	The team rarely talks about anything other than work.	Organize and Take part in team activities or discussions.
c+	/	/	Determine the process of the meeting, the discipline and the order in which the team speaks.
c+	/	/	Properly organize team activities such as dinners, informal discussions, etc.

Table20.issue1.3：Inefficient or even ineffective communication

type	factor	description	practical solutions/ envisaged solutions/practical precautions
a+	Are not familiar with each other or have differences	Some team members had accents that may lead to multiple repetitions in the discussion.	Ask to speak loudly and slow down speed of speech
a+	Team members have insufficient knowledge or different skill levels	Did not know much about the business-related content of the project of the project	Put forward problems in time and can check the knowledge document.
a+	Off topic in meeting Improper meeting process	The equipment needed for the meeting was not well prepared.	Do well in debugging the equipment in advance.
a+	Improper meeting process Off topic in meeting	At the beginning, the speaking order of the meeting was not fixed and there was a phenomenon of interrupting the speech	The unified meeting process is defined, and the speaking order and standing position are arranged.

a+	Off topic in meeting Improper meeting process	There was unnecessary discussion or descriptions in the meeting.	Stop arguments in a timely manner and record issues that arise, and control the rhythm to ensure that meetings do not deviate from the subject and proceed in an orderly manner.
a+	Lack of suitable communication tools	Sometimes the content needs to be repeated due to network fluctuations.	Speak slowly during meetings, maintain and check network equipment regularly.
a+	Team members have insufficient knowledge or different skill levels Difficult project knowledge sharing and synchronization	Did not know much about the business-related content of the project of the project	Establish and continuously update knowledge documents to share with the team in the form of text
b+	Off topic in meeting Improper meeting process	Did not make sufficient preparations for the content of the meeting before the review meeting.	Make a project checklist in advance.
b+	Difficult project knowledge sharing and synchronization	The issues discussed in the retrospective meetings will still appear in the next iteration	Pay attention to the summary of lessons learned after the meeting
b+	Off topic in meeting	Repeatedly mentioning repeated contents in retrospective meetings.	Prepare an outline before the meeting and control the rhythm.
b+	Improper meeting process	Far away from the microphone in the video conference made it not as clear as face-to-face	Ask speaker to speak loudly and slowly, and repeat as needed
b+	Differences in team members' understanding of knowledge Difficult project knowledge sharing and synchronization	It was difficult to unify opinions when encountering problems	Difficult team decisions are negotiated by voting
b+	Off topic in meeting Are not familiar with each other or have differences	Discipline in informal meetings is loose, and members are not interested in the content of the meeting	Team members exercise restraint and manage when needed.
c+			Prepare in advance what to say at the meeting

Table21. issue1.4：Misunderstanding in the process of communication

type	factor	description	practical solutions/ envisaged solutions/practical precautions
a+	Differences in team members' understanding of knowledge Difficult project knowledge sharing and synchronization	At the beginning, the team did not have a unified code writing standard,	Build a unified coding writing standard.
a+	Difficult project knowledge sharing and synchronization Team members fail to perform their personal duties	The developer ’s concealment of the project caused the project to slow down	Conduct the defect review of each code submission through agile software to confirm the progress
a+	Team members have insufficient knowledge or different skill levels Difficult project knowledge sharing and synchronization	Misunderstanding caused by incomprehension of business-related knowledge and failure to discover errors in knowledge documents in a timely manner	Communicate more with the Hong Kong team instead of relying solely on documents.
b+	Differences in team members' understanding of knowledge	Fail to fully understand the ideas expressed by others	Discuss a bit deeper and make the document more detailed
b+	Difficult project knowledge sharing and synchronization	Sometimes can only discuss issues with one person from other team and maybe the explanation is not detailed enough or the understanding is not thorough	Communicate and record the content of key issues using video conferencing
c+	/	/	Write as easy-to-understand documents as possible and share them in real time

From the analysis above, we did not find new issues related to communication. Seven new factors that could cause communication challenges were found by us(see table 17 for description). A total of 42 solutions were extracted and these solutions correspond to detailed communication issues one by one(See table 18-21 above for details).

5 DISCUSSION

In our study, we identified the most common issue encountered by distributed scrum teams by using system literature review, and based on the issue we found, we reviewed the selected articles to find parts of the factors that cause this issue. Then we used case study to find more issues related to communication, and found the corresponding factors and solutions to the communication issues.

5.1 Answers to RQs:

RQ1:What is the most common issue encountered by distributed scrum teams?
Answer to RQ1: From our analysis using SLR, the most common issue encountered by distributed scrum teams is "communication among stakeholders".

RQ2:What are the factors that cause these common issues?
Answer to RQ2: We totally found 16 factors that can cause the most common issue. Nine factors(F1-F9) can be seen in Table 13 which was found in SLR. And we found the other seven(F10-F16) are shown in Table 17. "Lack of suitable communication tools" and "Lack of trust or cohesion" were the main two factors that could lead to the communication issues according to the analysis of SLR. And "Off topic in meeting" was mentioned mostly by interviewees in case study.

RQ3:What are the solutions that can be used to solve these issues?
Answer to RQ3: We found in total 42 issue descriptions (which belong to the four detailed issues listed in Table 10 we classified in SLR) related to communication and corresponding solutions (practical solutions / envisaged solutions / practical precautions) after summarizing the interviewees' answers, which are all shown in the Table 18-21. Among the four communication challenges, the issue that interviewees encountered the most and gave the most opinions was "Some members do not participate in communication or have low enthusiasm for participation".

5.2 Research Findings

The following are the discoveries during our whole research process.

5.2.1 Discovery in SLR

1)People are very interested in distributed Scrum research. And the literature requires more specific industrial case studies and successful experience sharing to understand the details of using Scrum practices in globally distributed projects.

By reading various documents in SLR, we found that Scrum is one of the most popular methods in agile and has been applied in a distributed development environment according to Ghosh and Paasivaara et al.[4][9] Based on Paasivaara and Lous et al.[6][10],distributed Scrum has been receiving theoretical and applied research attention in both industry and academia, and there is growing interest in applying Scrum in distributed projects as mentioned by Hossain and Babar et al.[7][12] As we discovered in SLR, since 2007, there have been new articles to study distributed Scrum every year. Although there have been relevant research reports on the successful use of Scrum in distributed projects according to Szabó and Paasivaara et al. [3][5], few literatures can describe in detail how specific distributed teams deal with or prevent specific challenges in a specific environment.

In articles using Survey as a research method, because most of the people participating in the survey were from different companies, with very different Scrum development experience, work responsibilities, and different project backgrounds, their answers were basically all Based on their own work experience, the article did not describe their specific development environment and work background, which made us unable to know the background or type of team that the solution they mentioned were suitable for. For example, a survey by Dorairaj et al. [54] involves 45 agile

practitioners from 28 different software companies, and their agile development experience, project areas, iteration cycles, geographic distribution, and project duration were almost different.

In articles using Experiment as a research method， because most of the students who participated in the experiment did not have a lot of practical work and scrum experience, they mainly focus on learning rather than developing real business projects from the perspective of a real company's profit, so the problems they encountered and the corresponding solutions were not of great reference value to the real enterprises. For example, Damian et al.[2] mentioned that students from Finland and Canada participate in simulation of real software projects to complete the course of distributed software projects. Due to the inconsistency of teaching syllabus, the difference of curriculum activities and the limited project experience of students, they may need to participate in lectures to learn the knowledge of distributed project development or other activities,courses and could not devote themselves to project development. Therefore, there were some differences between their development process and the real business profit oriented enterprises.

In articles using SLR as a research method, many authors listed or described the issues mentioned in the articles obtained by the search and attach the solutions provided in the articles if there are any, but rarely categorize and classify all the issues in all articles systematically and discuss it in depth. In this way, we may be able to see a lot of similar descriptions of problems but we couldn't get the keypoint, and it was hard to understand the background or development environment from, so we may need to review the article to see the details of the solutions provided. For example, Esquivel[57] listed a lot of similar descriptions of issues found through SLR which we thought should be summarized or classified. Like he mentioned the description of "conference call does not work" several times. Although the author summarized these descriptions into several issues in the following, he did not give specific explanations for these issues. Besides, he did not correspond to the issues one by one when providing solutions.

Of all the articles using Case Study as a research method, As mentioned by Shrivastava, et al. [32], few articles focused on identifying and mitigating the most critical and common issues encountered in distributed scrum teams . For example, According to Hossain et al.[19], The author conducted four case studies and described the challenges encountered between them and the four cases, but did not make a quantitative calculation or suggest which challenge is the main problem. In our research, we not only identified the most common issue and conducted case study design around it, which brought our article focus.

2)Discovery of issues and factors

After extracting and summarizing the various research types of literature selected by us in SLR, we divided the discovered problems into three categories and counted them, as shown in the following Figure 20:

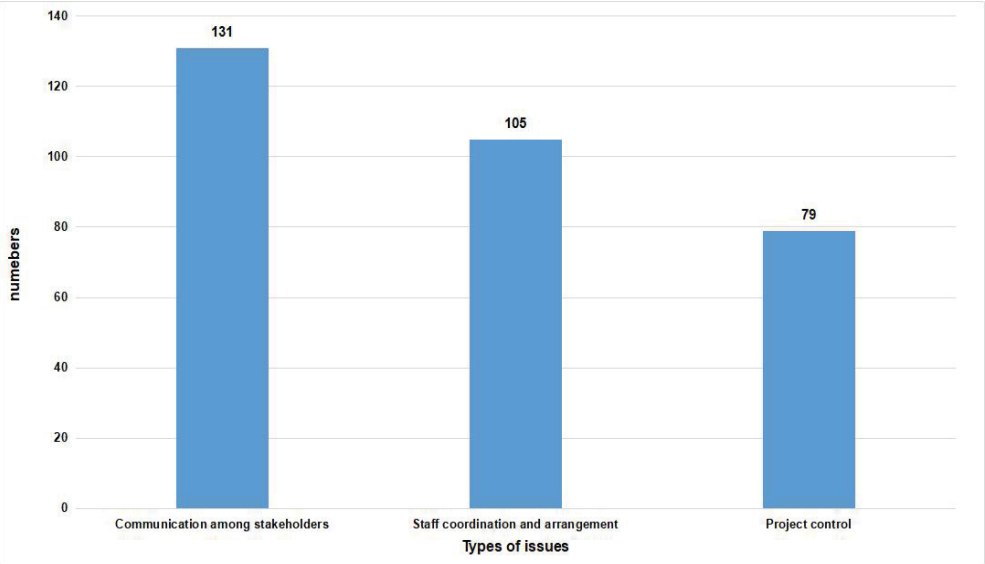

Figure 20.Types of issues found through SLR and their numbers

We could easily see that the number of the issue named “communication among stakeholders” was the biggest, and the numbers of the remaining two issues were still far away from it. So for the distributed scrum teams mentioned in these articles, “communication among stakeholders” was the most common issue.

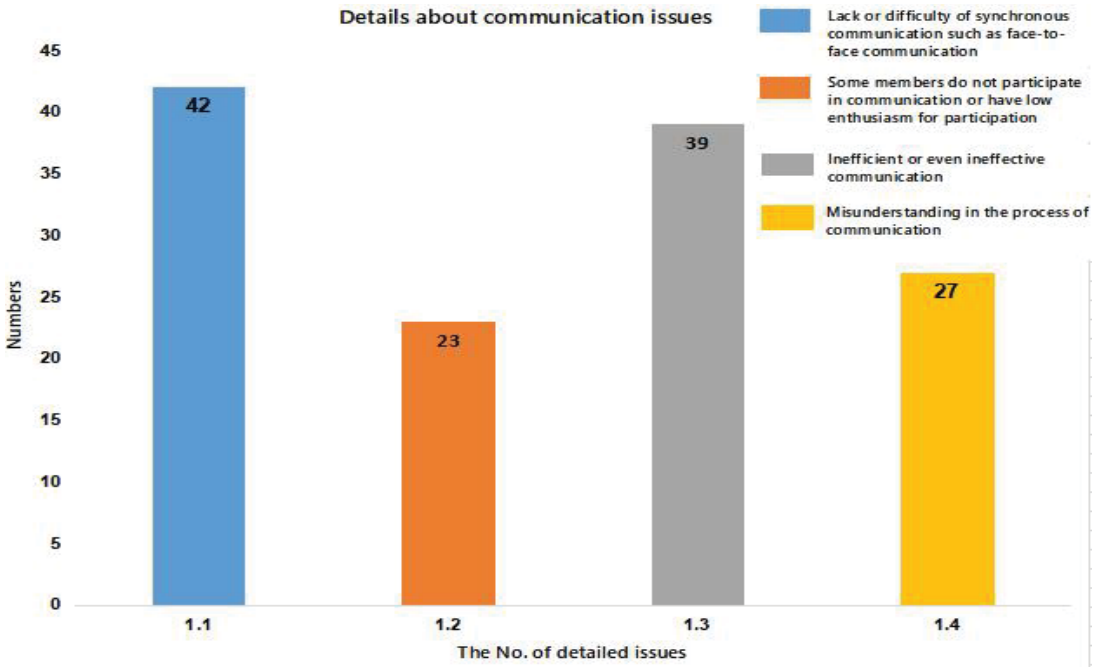

Figure 21.The detailed issues of Communication among stakeholders

As can be seen from the above Figure 21, in our SLR results, regarding "stakeholder communication", the two most frequent problems were "the lack or difficulty of face-to-face synchronous communication" and "inefficient or even effective communication". The remaining two problems occurred relatively few times.

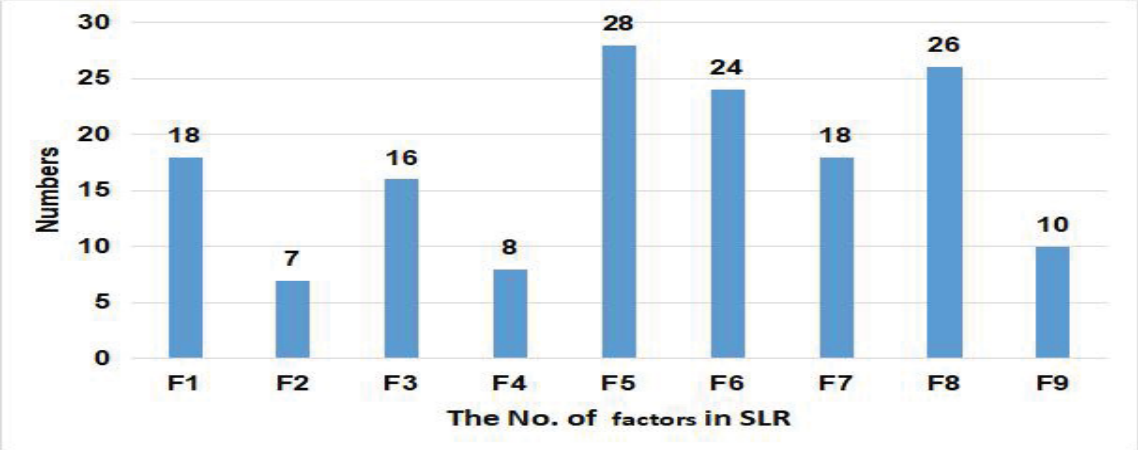

Figure 22.The factors of Communication among stakeholders

The above Figure 22 is the number of various factors that caused the challenge of "Communication among stakeholders" statistically calculated in the SLR. The names and detailed descriptions of each factor are in table 13 in section 4.1.3. In SLR, F5(Lack of suitable communication tools), F6(Are not familiar with each other or have differences) and F8(Lack of trust or cohesion) are all the factors mentioned many times in the literature that cause the distributed Scrum teams to encounter communication. Most of the members of these teams came from different countries or regions, they were not familiar with each other and might have differences in language, religion, politics, etc. And the lack of trust and cohesion between teams or among team members could also hinder their communication. Besides, suitable communication tools play an important role in communication. The smooth network, clear pictures, and high-quality call quality can greatly improve the efficiency and effectiveness of communication.

5.2.2 Discovery in case study

Based on the finding of SLR, we knew that for distributed scrum teams in articles, the most common issue was “communication among stakeholders”. And there were 4 detailed issues of communication and 9 factors could cause them. Then we found that in our case, everyone had some scrum project development experience, and everyone had used scrum in the distributed environment except one person. Besides, we had the following other findings.

The Figure 23 below is the interviewees' views on the challenges faced by distributed teams. We could see more than half of the interviewees mentioned communication which was also a big challenge for our case team.

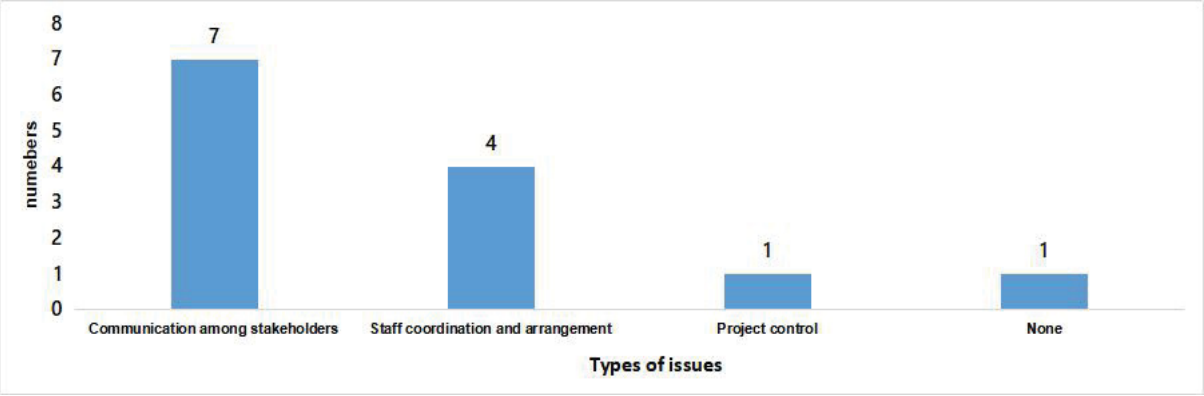

Figure 23. interviewees' views on the challenges faced by distributed teams

Through the interview, we failed to find new issues that related to communication. The issues mentioned by the interviewees could all be summarized into the four detailed issues of communication raised by our SLR.

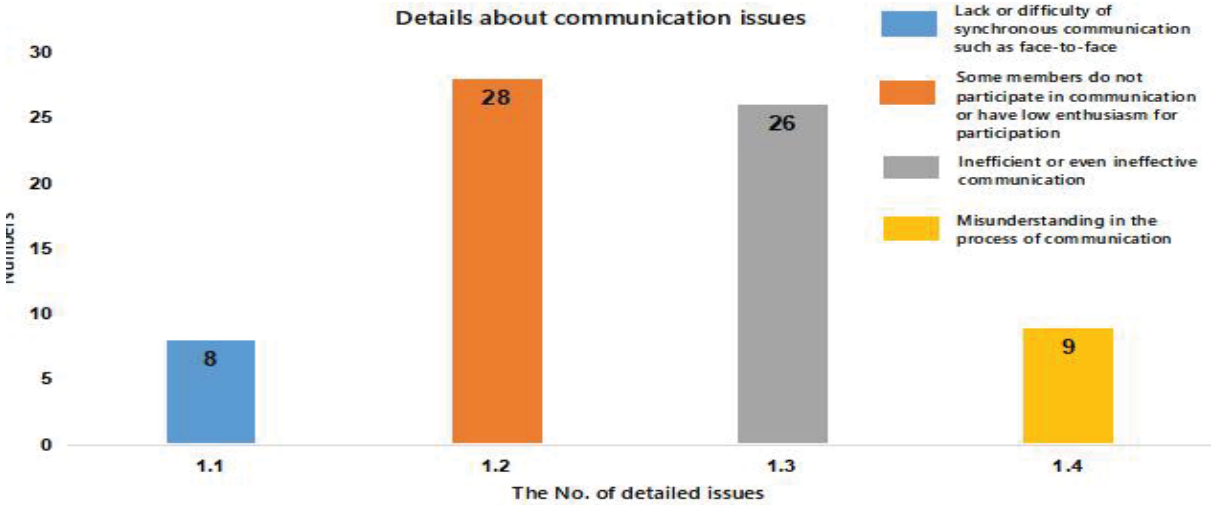

Figure 24.The detailed issues of Communication among stakeholders

As can be seen from the above Figure 24, in our case study results, the two most common challenges about "communication among stakeholders" are "some members do not participate in communication or have low enthusiasm for participation" and "inefficient or even in effective communication".

For the former, “are not familiar with each other or have differences” and “lack of trust or cohesion” are the main factors. The two teams in our case were working together for the first time, and they needed to spend a certain amount of time and effort on familiarizing each other and understanding each other's work styles. For detailed issues caused by these factors, our case team mainly used group activities, such as dinners or informal meetings, to enhance the team cohesion and understanding of each other.

For the latter, “off topic in meeting” and “improper meeting process” are the main factors. From this we found that the detailed issues caused by these factors basically occurred in the meetings. To solve the inefficiency in communication, the most important thing is to solve the waste of time caused by several meetings of scrum. After we sorted out the solutions, we concluded that the scrum master needs to maintain and debug the equipment in advance of the meeting, and formulate the meeting content and process in advance. Each member prepares for the meeting in advance and reduces discussions in the meeting that have nothing to do with the subject. The leader of the meeting should properly control the rhythm. These methods can effectively improve meeting efficiency.

It was worth noting that the above solutions were not suitable for all types of teams, but for the problems encountered by our case team, these solutions did play a certain helping role.

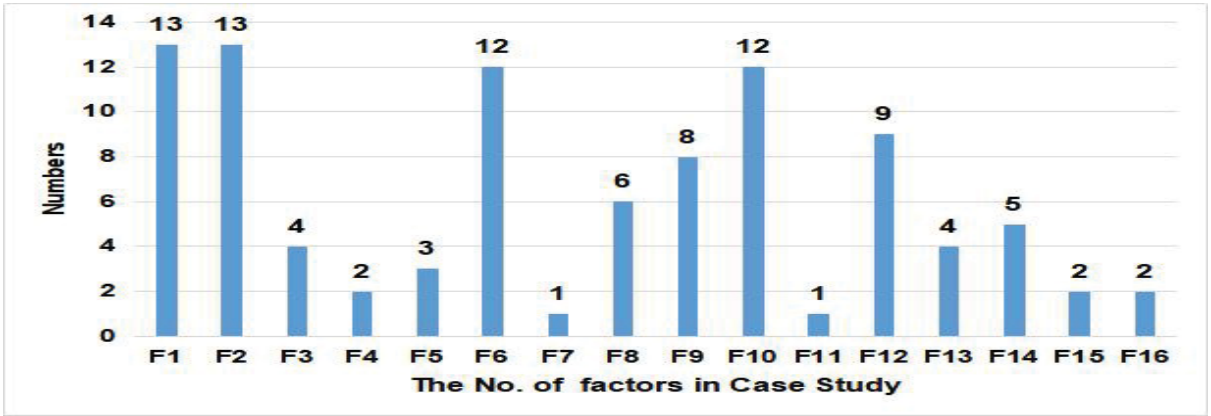

Figure 25. The factors of Communication among stakeholders

In the Figure 25 above, we counted the number of all the factors that caused the above 4 issues. The first nine factors are all factors obtained in our SLR. From F10 to F16 are the new factors we obtained from interviewees. From the interview results, we can see that the most mentioned factors are F1(Team members have insufficient knowledge or different skill levels) and F2(Difficult project knowledge sharing and synchronization). Then F6(Are not familiar with each other or have differences) and F10(Off topic in meeting) are tied in second place.

As for solutions, when we designed interview questions, we provided three types of answers for respondents to choose: a)Encountered before, b)Still experiencing, c)Never encountered. So we got different kinds of solutions from interviewees' answers, that is practical solutions, envisaged solutions and practical precautions. And these correspond to our a +, b +, and c + in table15 in Section 4.2. The practical solutions are the methods used by our case teams when they encountered the issue, and the issue was successfully solved. These have certain reference values for Scrum teams with the same development environment. As for envisaged solutions, these were proposed by the interviewees based on their own knowledge and experience which have not been put into practice. So we can use them as some suggestions for reference. Then practical precautions referred to that they have not been bothered by these problems. If we want to learn from these, it requires that the development environment and staff background of our teams should have a high degree of similarity to the teams in the case.

Although these interviewees were in the same team or even in the same geographic location and developed the same project , when asked the same question, they might give quite different answers. Take the daily stand-up meeting as an example, everyone participated in this meeting, but held different views on whether they had encountered problems in the meeting. Some interviewees thought that they had not encountered any problems in this meeting, while others had spoken out of the problems and might have given actual solutions. We considered this and thought this situation might be caused by the following reasons: 1.There are still some shortcomings in the interview questions designed by us. When we asked the questions, we did not try to give the interviewee a certain range, such as the beginning or the end of the project, so that the interviewee could not think of some details which were mentioned by other interviewees immediately when recalling. 2.Some of the issues that occurred were insignificant for some interviewees, these issues may not be particularly impressive for them so were not raised during the interview. 3. They might adopt different approaches or show different attitudes when they encounter the same issue because of their different academic qualifications, project experience and customs, etc. For example, with respect to the detailed issue of "inefficient or even ineffective communication", many of what the Hong Kong team mentioned was that the Shandong team did not understand the business module content and it took some time to explain, while the Shandong team mentioned the accent of the Hong Kong team members more. 4. Interviewees might be reluctant to mention the mistakes or problems they had made. And they might not want to say bad things behind others, so they did not disclose problems about others to us.

In addition to the above, interviewees sometimes gave irrelevant answers when asked. For example, HDP4 answered "Didn't go to the site to attend the review meeting to make our team members feel a lack of participation" when we asked him about "inefficient or even ineffective communication". This may be due to the fact that the interviewees were distracted or didn't understand our question, and we didn't notice it at that time.

With regard to the tool, we knew that the agile work software used by the case study team was TAPD. Almost every interviewee is satisfied with this tool. We summarize the advantages they mentioned during the interview as follows: It's free, online synchronization, open source and has a simple and friendly interface. And it has functional modules specifically for agile teams, can be used to record meetings, share project knowledge documents, find, record and share defects in the project, and it can be used to visualize the requirements, progress and completion of each member of the project. Therefore, we think that choosing a good and suitable tool is very important for distributed software development teams, which can help them communicate better and understand each other's progress and status, so as to improving the efficiency and quality of the project.

5.2.3 Summary of results

1)In SLR, we identified three types of challenges encountered by distributed scrum teams (1. Communication among stakeholders, 2. Staff coordination and arrangement, 3. Project control). We have listed out 4 main issues(see the details in Table 10 in Section 4.1.3) related to "1. Communication among stakeholders". In Case Study, we did not find new issues about communication.The proportion of these four major issues in the SLR and Case Study results are shown in the figures below.

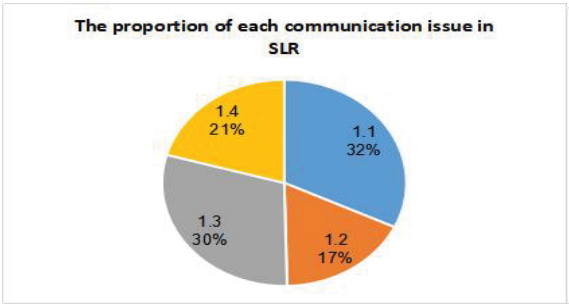

Figure 26. Detailed issues in SLR

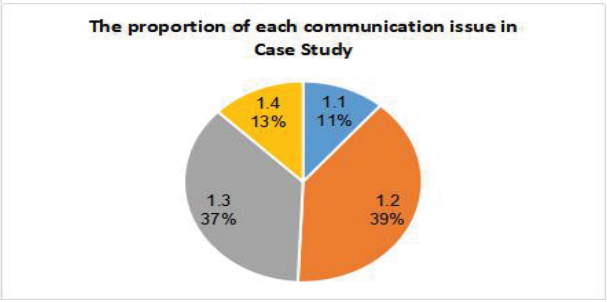

Figure 27. Detailed issues in case study

As shown in the above figure, compared with the SLR results, in case study, issue 1.1 "Lack or difficulty of synchronous communication such as face-to-face" is significantly reduced, while issue 1.2 "Some members do not participate in communication or have low enthusiasm for participation" has greatly increased. Based on the results of our interviews, we believe that this may be because the team conducting the case analysis did not cross the time difference and the geographical time distance is not long, so the number of people incapable of face-to-face communication is less. But the cultural distance between the two teams led to the increase of the second problem. And since both teams are from the same country / region, the language is the same, except that the accent affects but basically does not affect the communication. So issue 1.4 "Misunderstanding in the process of communication" rarely happens. Misunderstanding is mainly due to different levels of knowledge. Regardless of SLR results or Case Study results, issue 1.3 "Inefficient or even in effective communication" has a high proportion. This shows how to improve the efficiency of communication in distributed scrum development is an important issue.

2)After identifying the most common issue, we reviewed the articles we used in SLR and found 9 factors that caused this issue——"communication among stakeholders". Then we obtained another 7 factors through case study. We noticed in the interview that all the factors we discovered in the SLR were mentioned by the interviewees. The percentages of each factor found in SLR and case study are shown in the figures below:

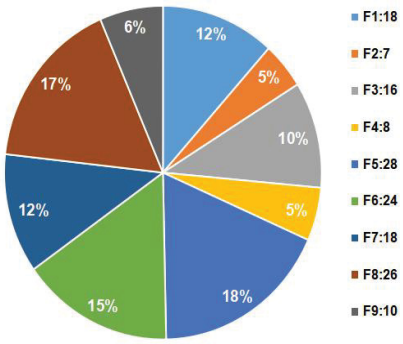

Figure 28.Factors ratio chart in SLR

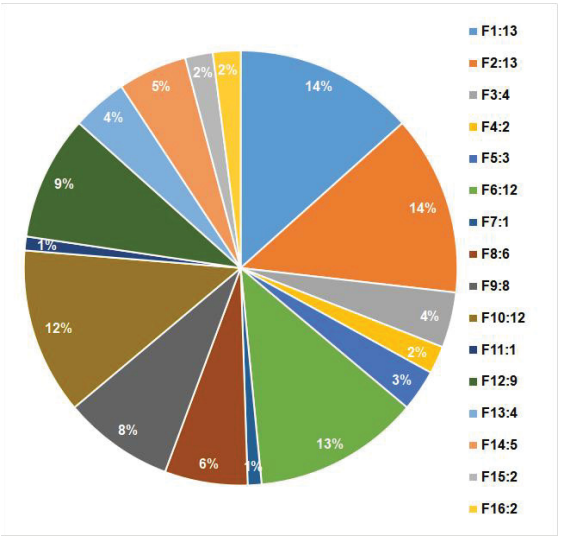

Figure 29.Factors ratio chart in Case Study

In the two pie charts above, we could see that the factors we found in SLR were all mentioned by interviewees in the case study(the interviewees 'descriptions of these factors were very close, so we grouped them together).

In both SLR and case study, F1(Team members have insufficient knowledge or different skill levels) and F6(Are not familiar with each other or have differences) are the two main factors that cause the communication challenges in the distributed teams. It was worth noting that F7(Unavailability of stakeholders or failing to respond in a timely manner) was mentioned many times in the SLR and only once in the case study. Since our teams were in the same country, and the customer as well as development team were also in the same city, there was no time difference and no large geographic distance, so there was no delay in communication unless the customer was sometimes busy with other private things.

We found 7 other factors that can lead to communication challenges in the case study. They mainly came from the interviewees' answers to the questions about the meetings in Scrum. Because the articles we selected in SLR rarely have an in-depth discussion on the Scrum meetings. What they mentioned were just descriptions of the scrum meetings and didn't explore the causes of communication issues at the meeting. But we specially set up interview questions for the Scrum meeting, asking the interviewees to recall and answer the issues existing in the meetings, so we thought this was the important reason why these new factors came into being.

Besides, data induction of the challenges encountered by various distributed scrum teams in SLR was not very representative because the distributed scrum teams mentioned in SLR had different development environments, personnel background, distribution, etc. Like the teams in SLR, the teams in our case also encountered the same challenges related to communication. But What was different from most of the teams mentioned in SLR was that the members of our case team were all from the same country, there was no great geographical distance and no time difference, and they spoke the same language except for the presence of some accents. So compared with the general or comprehensive data obtained in our SLR, the results of the interview in our case teams had some more reference value relatively.

3)When proposing solutions to specific issues, analyze the issue environment and the factors at the same time.
In whether SLR or case study, we found that there were many factors that could cause communication issues in the distributed scrum teams. One issue might be caused by several factors, and one factor could also cause several different issues. So we needed to analyze the specific situations when we proposed solutions, because no one solution was omnipotent and could not be used to solve all issues caused by different factors in all scenarios. For example, if the communication efficiency was low during the daily standing meeting, we needed to know what specific factors caused it, whether it was because the team members did not understand the meaning expressed by the other party or because the video call quality was not good, etc. Therefore, it was necessary to analyze the specific issues and specific factors in a specific environment before we proposed a suitable solution.

4)In the above related work, according to Bjarnason et al.[123] , the challenges of the distributed software development team they studied were coordination and communication. According to their research, they defined a total of eight distances that could affect communication and coordination of distributed software development teams. They also provided eight abstract practices, each of which could act (extend or shorten) some of the above mentioned distances to help development organizations better deal with the challenges of communication and coordination in the development process. However, our research aims to find out the challenges, the factors and solutions of the distributed scrum development team in the development process. After a systematic literature review, we identified that the most common challenge was communication, and carried out the follow-up research on communication. Their article does not distinguish communication and coordination clearly, but regarded them as one and carried out research. With respect to factor, they mentioned the geographical, temporal, cultural distance, and we treated them as DSD properties and explained them

in detail in section 2.1.1. These three findings were similar to ours. Then, except D3, D4, other factors were not explicitly proposed in our research, because our research was mainly aimed at the challenges of communication, and their paper was aimed at communication and coordination, but the specific issues related to coordination and project control summarized in section 4.1.3 had certain relationship with the remaining factors mentioned in their article. So we thought if we continued to study coordination and project control in the future, it was likely to find similar factors to them. In addition, scrum, as an implementation of agile, the 4 defined meetings had made some changes to the way and process of communication between teams. And we also set up a series of questions related to scrum meeting in the interview, some of the factors we got and solutions were related to scrum meeting. This also meant that some of the challenges faced by the distributed scrum development team about communication were related to scrum. Of course, we could not say that scrum framework was problematic. After all, it was fixed, and how to better apply it in the development process was more important. From the macro perspective, our research and their article both mentioned the communication challenge faced by distributed teams. But from a deeper perspective, we also made some research and discussion on how to apply scrum framework to the case team, and found that the development team also encountered some problems when applying scrum, which affected the communication of the teams. So we think that the good or bad application of scrum has a real impact on the communication of distributed software development team.

5.3 Validity threats

The following are the validity threats in our research process.

5.3.1 Validity threats in SLR

The main threats to this SLR process are the review protocol, paper selection, and data extraction. The following contents will further illustrate these threats.

This systematic literature review process and the detailed review protocol was designed before conducting the review. Since we did not have much experience in performing SLR, we studied three guidelines and tried to get some ideas from the processes and review protocols they listed by Kitchenham and Wright et al.[113][114][115] After comparison and discussion, we finally selected article[113] which was written by Kitchenham et al.[113] as our main source of guidance and consulted mostly from it based on our subjective thoughts.

According to Kitchenham et al.[113], bias in paper selection could be a result of publication bias. Publication bias refers to the issue where positive results are more likely to be published than negative results.In our article, although this risk existed, it did not have a large impact on our whole. Because in our article we didn't pursue the comprehensiveness and effectiveness of all results. For this research, we were most interested in finding the most common issues in the distributed scrum team, so it was more important for us to be able to achieve the widest possible coverage of information resources. On this basis, the reliability of the article is guaranteed.

We consulted many papers to improve our review protocols, so as to obtain more consistent and accurate results. But it was still possible that we missed some research that was related to our topic. The 96 search strings we used helped us to get thousands of papers, but many of them were duplicated and it was hard for us to scan each paper in detail. Besides, the number of databases we selected and the resources they could find were limited. Although we obtained several papers from other databases through performing snowball sampling, there was a chance to miss many relevant papers from other databases.

In this systematic literature review, a large number of articles were deleted by us and they were considered irrelevant to the subject during our inclusion and exclusion process. So there is a risk of excluding meaningful articles. So in order to reduce this risk, we first discussed with experienced supervisors when determining the design and evaluation of the literature review and let them review and monitor our process. When we determined the initial inclusion and exclusion criteria, the two researchers met to discuss until the criteria reached the same understanding. Each researcher kept a list

of all excluded articles for the second review by another researcher after each iteration of the literature that he was responsible for. It also marked the discussion of articles with different opinions. In addition, we adopted the method proposed by Ivarsson and Gorschek [94] to evaluate rigor and relevance of each paper in quality assessment. The results of the assessment were largely related to our subjective views and therefore it also brought potential threats. Finally, because the length of some articles was too long or the typesetting was not clear, it increased our reading difficulty, which would also cause us to miss some obvious challenges during data extraction and affected the final statistical results.

5.3.2 Validity threats in case study

Because the number of the interview questions we designed and interview time were limited, the interviewee might not be able to think of all the information at the first time during the interview, and although we could help the interviewee to open his mind due to our design of asking questions when asking for information about the scrum meetings, it also caused the interviewees to mention some things that were irrelevant to our research topic. The guidance work in the process of interviewing about scrum meeting questions was not particularly satisfying. When we asked questions, we did not try to provide some scope for the interviewees, such as at the beginning or the end of the project, this may also cause interviewees to be unable to recall something quickly that happened at a particular point in time and just answered “nothing”. Now we thought we could have asked them like this at that time: what impressed you most at the daily standing meeting in the early stage of project development.

The case team we interviewed has its own particularity, like no time difference and have the same native language, etc. The interview results of them were only applicable to the same type of distributed scrum teams as them, but not to all distributed scrum teams, so the representativeness of this case is very limited. More various case teams need to be interviewed to supplement the information.

Finally, since the induction of this case study was not statistical but analytical, the induction had a certain degree of randomness and subjectivity. We could not guarantee that we fully understood and correctly summarized the answers provided by the interviewees. Although we have two researchers for analysis and statistics, this could not completely eliminate the subjectivity and misunderstandings in the statistical results.

6 CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

1.Through our above reading and analysis of various literatures, we found that people still had strong interests in the challenges encountered by distributed Scrum teams and how to solve these challenges. People hoped that they could find suitable solutions from the literature to try to mitigate, solve or avoid the challenges. And although some related articles had mentioned successful cases of distributed Scrum team development projects, these articles were more about successful experience sharing and the unique measures they adopted. For the background introduction of the case teams, the challenges they encountered and how to solve them were not very detailed, so it would be difficult for other development teams to judge whether the measures mentioned in the articles apply to themselves. There was still a great demand for articles on research on specific industrial cases, especially articles that conducted in-depth research on a specific issue in a specific case team. In order to make the mentioned cases have reference significance, more descriptions of the team and the background of the project should be provided, and specific descriptions of the challenges encountered should be taken, the factors also needed to be described. And then detailed solutions should be given one by one.

2.In our research, we found challenges of a distributed scrum team mentioned in all articles through reading literature and divided them into three categories: communication among stakeholders, staff coordination and arrangement, project control. Then we counted the number of times these challenges were mentioned.

Table 22. Challenges found and counts

Type of challenges	Total number
Communication among stakeholders	131
Staff coordination and arrangement	105
Project control	79

With the support of the above data, we determined that “communication among stakeholders” was the most common challenge. Its detailed issues are shown in the table below. And the descriptions of these issues could be found in section 4.1.3.

Table 23.The detailed issues of communication

1.1 Lack or difficulty of synchronous communication such as face-to-face communication
1.2 Some members do not participate in communication or have low enthusiasm for participation
1.3 Inefficient or even ineffective communication
1.4 Misunderstanding in the process of communication

Combining the results of SLR and Case study, we found that a total of 16 factors could cause the above communication issues. The 16 factors were shown in the table below. And the descriptions of these issues could be found in Table 13 in section 4.1.3 and Table 17 in section 4.2.

Table 24.The factors leading to communication challenges

F1.Team members have insufficient knowledge or different skill levels

F2.Difficult project knowledge sharing and synchronization
F3.Differences in team members' understanding of knowledge
F4.Travel costs are too high
F5.Lack of suitable communication tools
F6.Are not familiar with each other or have differences
F7.Unavailability of stakeholders or failed to respond in a timely manner
F8.Lack of trust or cohesion
F9.Team members do not speak or speak very little during the communication process
F10.Off topic in meeting
F11.policy
F12.Improper meeting process
F13.Team members fail to perform their personal duties
F14.Team members lack a sense of participation
F15.Different work practice standards or work habits
F16.Tense working atmosphere

In total, we summarized 42 solutions, of which 19 were actually adopted and solved the communication challenges encountered by the case teams, 18 were conceived by the interviewee but not put into practice, and the remaining 5 were used by case teams to avoid the corresponding challenges we asked in the interview.

6.2 FUTURE WORK

1. In our SLR, we searched more than 3,000 articles through 5 digital databases and snowball sampling, but there were a lot of repeated articles and only 60 articles were finally adopted by us. Although the tool EndNote helped us a lot, it still took us a lot of time and energy to search and organize these articles. And many of the searched articles may have nothing to do with our research topics or did not meet our quality assessment standards.We thought that in the future we could improve our search keywords by reducing their number. In fact, although these keywords were different, they had the same meaning which would cause us to get a large number of repeated articles in the different literature databases. When searching for a certain number of documents that met our requirements, we should use more snowballing sampling strategies, which could help us quickly find more articles that were relevant to our topic without having to screen one by one from plenty of articles. In addition, we should increase the search database in the future, because there may be many articles related to our topic in other databases.

2. In our analysis of our case study results, we found that the proportion of the number of communication problems we got was very different from the SLR results. And in our case study, we also discovered some factors that caused these communication problems that were not counted in SLR.We believed that the reason for this situation is that in the SLR, we did not organize according to the type of distributed scrum teams, but count all types of teams.This means that our solutions are only valid for teams of the type we have defined. We do not claim that our findings are common to all

distributed agile projects. Therefore, in the future research, the case study team can be defined in detail, and then the case analysis can be carried out many times to make the research more universal.

3. From our perspectives,whether or not the problem was solved was a subjective opinion. In our case, there was a challenge about understanding the content of business modules. The Hong Kong team said it was solved because they thought they provided the knowledge document, while the Shandong team thought it was not solved yet because they still couldn't fully understand it and needed more communication . This meant that some team members may think that the solution was effective, while others may think that the solution had not achieved the desired results. Therefore, in our interview, it appeared that different people in the same team held different attitudes or views on the same challenge.From the above, we believed that we should pay attention to the relevance of the interviewees' answers during the interview. When asking the interviewee, we could appropriately introduce the views of other interviewees, which could help the interviewee think more widely and we could also find the reasons for the differences in views or attitudes.

4. We think that in our interview, when the issues described by the two people are the same, some people think that the solutions they have taken are effective, and successfully reduced the impact of the problem, but some people think that even if the solution is provided but the issue still exists. This also reflects a problem: some solutions are not necessarily effective for everyone. In other words, some solutions did not completely solve the problem, but can only achieve the effect of mitigation.
Our strategy for this situation is to sort out this solution when it is useful to some people, and also incorporate it into our solution. In future research, we can feed back the opinions of those who think the solution is not useful to those who think the solution is useful, find out the causes of the differences and try to improve these solutions to benefit more people.

5. In our research, we just looked for and focused on the communication-related challenges encountered in distributed scrum development and their solutions. There are still many challenges in areas such as team coordination and project control as we found in SLR. In our future work, we think we can continue our in-depth research, study about other difficulties and challenges that may be encountered in distributed scrum, and summarize solutions to these challenges.

6. In the future, more verification methods may be needed to check the effectiveness of the solution. The solutions proposed in our research have a positive effect on our case team. We hope to implement these solutions on other similar kinds of teams and verify whether they are effective or helpful to project development.

7 REFERENCES

[1] Gilberto Borrego, Alberto L Morán, Ramón René Palacio Cinco, Oscar Mario Rodríguez-Elias, and Eloísa García-Canseco. Review of approaches to manage architectural knowl-edge in agile global software development.IET Software, 11(3):77–88, 2017.

[2] Daniela Damian, Casper Lassenius, Maria Paasivaara, Arber Borici, and Adrian Schröter.Teaching a globally distributed project course using scrum practices. In 2012 Second International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD), pages 30–34. IEEE, 2012.

[3] Dávid Marcell Szabó and Jan-Philipp Steghöfer. Coping strategies for temporal, geographical and socio cultural distances in agile gsd: a case study. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), pages 161–170. IEEE, 2019.

[4] Gopal K Ghosh. Challenges in distributed scrum.In2012 IEEE Seventh International Conference on Global Software Engineering, pages 200–200. IEEE, 2012.

[5] Maria Paasivaara, Sandra Durasiewicz, and Casper Lassenius. Distributed agile development:Using scrum in a large project. In 2008 IEEE International Conference on Global Software Engineering, pages 87–95. IEEE, 2008.

[6] Pernille Lous, Marco Kuhrmann, and Paolo Tell. Is scrum fit for global software engineering? In2017 IEEE 12th International Conference on Global Software Engineering(IGCSE), pages 1–10. IEEE, 2017.

[7] Emam Hossain, Muhammad Ali Babar, and Hye-young Paik. Using scrum in global software development: a systematic literature review. In 2009 Fourth IEEE International Conference on Global Software Engineering, pages 175–184. Ieee, 2009.

[8] Maria Paasivaara and Casper Lassenius. Scaling scrum in a large distributed project.In 2011 International Symposium on Empirical Software Engineering and Measurement,pages 363–367. IEEE, 2011.

[9] Maria Paasivaara, Casper Lassenius, and Ville T Heikkilä. Inter-team coordination in large-scale globally distributed scrum: Do scrum-of-scrums really work? In Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement,pages 235–238, 2012

[10] Maria Paasivaara, Sandra Durasiewicz, and Casper Lassenius. Using scrum in distributed agile development: A multiple case study. In 2009 Fourth IEEE International Conference On Global Software Engineering, pages 195–204. IEEE, 2009.

[11] Mauricio Cristal, Daniel Wildt, and Rafael Prikladnicki. Usage of scrum practices within a global company. In 2008 IEEE International Conference on Global Software Engineering,pages 222–226. IEEE, 2008.

[12] Emam Hossain, Muhammad Ali Babar, Hye-young Paik, and June Verner. Risk identification and mitigation processes for using scrum in global software development: A conceptual framework. In2009 16th Asia-Pacific Software Engineering Conference, pages 457–464. IEEE, 2009.

[13] Rajeev Kumar Gupta and Prabhulinga Manikreddy.Challenges in adapting scrum in legacy global configurator project. In 2015 IEEE 10th International Conference on Global Software Engineering, pages 46–50. IEEE, 2015.

[14] Mira Kajko-Mattsson, Gayane Azizyan, and Miganoush Katrin Magarian. Classes of distributed agile development problems. In 2010 Agile Conference, pages 51 – 58. IEEE,2010.

[15] Youry Khmelevsky, Xitong Li, and Stuart Madnick. Software development using agile

and scrum in distributed teams. In2017 Annual IEEE International Systems Conference(SysCon), pages 1–4. IEEE, 2017.

[16] Elaine Therrien. Overcoming the challenges of building a distributed agile organization.In Agile 2008 Conference, pages 368–372. IEEE, 2008.

[17] Murat Dogus Kahya and Çagla Seneler. Geographical distance challenges in distributed agile software development: Case study of a global company. In 2018 3rd International Conference on Computer Science and Engineering (UBMK), pages 78 – 83. IEEE, 2018.

[18] Areebah Altaf, Urooj Fatima, Wasi Haider Butt, Muhammad Waseem Anwar, and Maryum Hamdani. A systematic literature review on factors impacting agile adaptation in global software development. In Proceedings of the 2019 7th International Conference On Computer and Communications Management, pages 158–163, 2019.

[19] Emam Hossain, Paul L Bannerman, and Ross Jeffery. Towards an understanding of tailor-ing scrum in global software development: a multi-case study. InProceedings of the 2011 International Conference on Software and Systems Process, pages 110–119, 2011.

[20] Maria Paasivaara, Casper Lassenius, Daniela Damian, Petteri Räty, and Adrian Schröter.Teaching students global software engineering skills using distributed scrum. In the 201335th International Conference on Software Engineering (ICSE), pages 1128 – 1137.IEEE,2013.

[21] Ingo Richter, Florian Raith, and Michael Weber. Problems in agile global software engineering projects especially within traditionally organised corporations: [an exploratory semi-structured interview study]. InProceedings of the Ninth International C* Conferenceon Computer Science & Software Engineering, pages 33–43, 2016.

[22] Mohammad Shameem,Rakesh Ranjan Kumar, Chiranjeev Kumar, Bibhas Chandra, andArif Ali Khan. Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process.Journal of Software: Evolution and Process, 30(11): e1979, 2018.

[23] Buturab Rizvi, Ebrahim Bagheri, and Dragan Gasevic. A systematic review of distributed agile software engineering.Journal of Software:Evolution and Process, 27(10):723–762, 2015.

[24] Shahla Ghobadi and Lars Mathiassen. Risks to effective knowledge sharing in agile software teams:a model for assessing and mitigating risks.Information Systems Journal,27(6):699–731, 2017.

[25] Maria Paasivaara, Sandra Durasiewicz, and Casper Lassenius. Using scrum in a globally distributed project: a case study.Software Process: Improvement and Practice, 13(6):527–544, 2008.

[26] Syeda Sumbul Hossain. Challenges and mitigation strategies in reusing requirements in large-scale distributed agile software development: A survey result. Intelligent Computing-Proceedings of the Computing Conference, pages 920–935. Springer, 2019.

[27] Wasim Alsaqaf, Maya Daneva, and Roel Wieringa. Quality requirements challenges in the context of large-scale distributed agile: An empirical study.Information and software technology, 110:39–55, 2019.

[28] Wasim Alsaqaf, Maya Daneva, and Roel Wieringa. Quality requirements in large-scale distributed agile projects–a systematic literature review. International Working Conference on Requirements Engineering: Foundation for Software Quality, pages 219 – 234.Springer, 2017.

[29] Seiyoung Lee and Hwan-Seung Yong. Distributed agile: project management in a global environment.Empirical Software Engineering, 15(2):204–217, 2010.

[30] Mark Rajpal. Lessons learned from a failed attempt at distributed agile. InInternationalConference on Agile Software Development, pages 235 – 243. Springer, Cham, 2016.

[31] Maria Paasivaara, Benjamin Behm, Casper Lassenius, and Minna Hallikainen.Large-scale agile transformation at ericsson: a case study.Empirical Software Engineering,23(5):2550–2596, 2018.

[32] Suprika V Shrivastava and Urvashi Rathod. Categorization of risk factors for distributed agile projects.Information and Software Technology, 58:373–387, 2015.

[33] VN Vithana, David Asirvatham, and Md Gapar Md Johar.Investigating the issues of using agile methods in offshore software development in sri lanka. In Asian Conference On Intelligent Information and Database Systems, pages 515–523. Springer, 2017.

[34] Daniela S Cruzes, Nils B Moe, and Tore Dybå. Communication between developers and testers in distributed continuous agile testing.In 2016 IEEE 11th International Conference On Global Software Engineering (ICGSE), pages 59–68. IEEE, 2016.

[35] Outi Sievi-Korte, Kari Systä, and Rune Hjelsvold. Global vs. local—experiences from a distributed software project course using agile methodologies. In2015 IEEE Frontiers in Education Conference (FIE), pages 1–8. IEEE, 2015.

[36] Sarah Beecham,John Noll, and Ita Richardson. Using agile practices to solve global software development problems – a case study. In 2014 IEEE International Conference on Global Software Engineering Workshops, pages 5–10. IEEE, 2014.

[37] Abbas Moshref Razavi and Rodina Ahmad. Agile development in large and distributed environments: A systematic literature review on organizational, managerial and cultural aspects. In2014 8th. Malaysian Software Engineering Conference (MySEC), pages 216–221. IEEE, 2014.

[38] Paul L Bannerman, Emam Hossain, and Ross Jeffery. Scrum practice mitigation of global software development coordination challenges: A distinctive advantage? In 2012 45th Hawaii International Conference on System Sciences, pages 5309–5318. IEEE, 2012.

[39] Tatiane Lautert, Adolfo Gustavo Serra Seca Neto, and Nádia P Kozievitch. A survey on agile practices and challenges of a global software development team. In Brazilian Workshop on Agile Methods, pages 128–143. Springer, 2019.

[40] Mohammad Shameem, Bibhas Chandra, Rakesh Ranjan Kumar, and Chiranjeev Kumar. A systematic literature review to identify human related challenges in globally distributed agile software development: towards a hypothetical model for scaling agile methodologies. In 2018 4th International Conference on Computing Communication and Automation (ICCCA), pages 1–7. IEEE, 2018.

[41] Nils Brede Moe, Daniela S Cruzes, Tore Dybå, and Ellen Engebretsen. Coaching a global agile virtual team. In 2015 IEEE 10th International Conference on Global Software Engineering, pages 33–37. IEEE, 2015.

[42] Amitoj Singh, Kawaljeet Singh, and Neeraj Sharma. Agile in global software engineering: an exploratory experience. International Journal of Agile Systems and Management, 8(1):23–38, 2015.

[43] Mohammad Abdur Razzak and Rajib Ahmed. Knowledge sharing in distributed agile projects: Techniques, strategies and challenges. In2014 Federated Conference on Computer Science and Information Systems, pages 1431–1440. IEEE, 2014.

[44] Biyagamage Agra Junius Fernando, Tracy Hall, and Anthony Fitzpatrick. The impact of media selection on stakeholder communication in agile global software development: apreliminary industrial case study. In Proceedings of the 49th SIGMIS annual conference on Computer personnel research, pages 131–139, 2011.

[45] Steven Fraser,Pekka Abrahamsson, Robert Biddle, Jutta Eckstein,Philippe Kruchten,Dennis Mancl, and Werner Wild. Culture and agile: Challenges and synergies. International Conference on Agile Processes and Extreme Programming in Software Engineering,pages 251–255. Springer, 2008.

[46] Areej Al-Zaidi and Rizwan Qureshi. Global software development geographical distance communication challenges.Int. Arab J. Inf. Technol., 14(2):215–222, 2017.

[47] Kevin Dullemond, Ben van Gameren, and Rini van Solingen. How technological support can enable advantages of agile software development in a gse setting. In 2009 Fourth IEEE International Conference on Global Software Engineering, pages 143–152. IEEE,2009.

[48] Imran Ghani, Angelica Lim, Muhammad Hasnain, Israr Ghani, and Muhammad Imran Babar. Challenges in distributed agile software development environment: A systematic literature review.KSII Transactions on Internet & Information Systems, 13(9), 2019.

[49] Muhammad Usman, Qamar Abbas, Beebarg Akram, and Junaid Hussain. Diminution of issues and challenges when using scrum in global software engineering.International Journal of Technology and Research, 4(2):31–43, 2016.

[50] Md Rahman, Arijit Das, et al. Mitigation approaches for common issues and challenges when using scrum in global software development, 2015.

[51] Suprika Vasudeva Shrivastava and Hema Date. A framework for risk management in globally distributed agile software development (agile gsd).differences, 4:13, 2010.

[52] Islam Zada and Sara Shahzad. Issues and implications of scrum on global software development.Bahria University Journal of Information & Communication Technologies(BUJICT), 8(1), 2015.

[53] Abdulaziz Alsahli, Hameed Khan, and Sultan Alyahya. Agile development overcomes gsd challenges: A systematic literature review.International Journal of Computer Science andSoftware Engineering, 6(1):7, 2017.

[54] Siva Dorairaj, James Noble, and Petra Malik. Knowledge management in distributed agile software development. In2012 Agile Conference, pages 64–73. IEEE, 2012.

[55] Siva Dorairaj, James Noble, and Petra Malik. Understanding lack of trust in distributed agile teams: A grounded theory study. 2012.

[56] Juyun Cho. Distributed scrum for large-scale and mission-critical projects.AMCIS 2007Proceedings, page 235, 2007.

[57] Salvador Esquivel. Communication issues in agile software development. InXXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)., 2016.

[58] Pawanpreet Kaur and Sumit Sharma. Agile software development in global software en-gineering.International Journal of Computer Applications, 97(4), 2014.

[59] Mohammad Emam Hossain.Scrum Practice Mitigation of Coordination Challenges in Global Software Development Projects: An Empirical Study. PhD thesis, Ph. D. Thesis,The University of New South Wales Sydney 2052, Australia, 2011.

[60] Yehia Ibrahim Alzoubi, Asif Qumer Gill, and Ahmed Al-Ani. Empirical studies of geographically distributed agile development communication challenges: A systematic review.Information & Management, 53(1):22–37, 2016.

[61] Suprika Vasudeva Shrivastava et al. Distributed agile software development: A review.arXiv preprint arXiv:1006.1955, 2010.

[62] Erran Carmel.Global software teams:collaborating across borders and time zones. Prentice Hall PTR, 1999.

[63] Rafael Prikladnicki, Jorge Luis Nicolas Audy, and Roberto Evaristo. A reference model for global software development: findings from a case study. In 2006 IEEE International Conference on Global Software Engineering (ICGSE'06), pages 18–28. IEEE, 2006.

[64] Rafael Prikladnicki, Jorge Luis N Audy, Daniela Damian, and Toacy C de Oliveira. Distributed software development: Practices and challenges in different business strategies of offshoring and onshoring. International Conference on Global Software Engineering(ICGSE 2007), pages 262–274. IEEE, 2007.

[65] Lucas Layman, Laurie Williams, Daniela Damian, and Hynek Bures. Essential communication practices for extreme programming in a global software development team.Information and software technology, 48(9): 781–794, 2006.

[66] Sundeep Sahay. Global software alliances: the challenge of standardization'.Scandinavian Journal of Information Systems, 15(1):11, 2003.

[67] Samireh Jalali and Claes Wohlin. Global software engineering and agile practices: a systematic review. Journal of software: Evolution and Process, 24(6): 643–659, 2012.

[68] Helena Holmström, Brian Fitzgerald, Pär J Ågerfalk, Eoin Ó Conchúir, et al. Agile prac-tices reduce distance in global software development.Information systems management, 23(3):7–18, 2006.

[69] Par J Agerfalk, Brian Fitzgerald, Helena Holmstrom Olsson, Brian Lings, Bjorn Lundell, and Eoin Ó Conchúir. A framework for considering opportunities and threats in distributed software development. 2005.

[70] Madan Mohan Jha, Rosa Maria Ferrer Vilardell, and Jai Narayan. Scaling agile scrum software development: providing agility and quality to platform development by reducing time to market. In 2016 IEEE 11th international conference on global software engineering (ICGSE), pages 84–88. IEEE, 2016.

[71] Tulasi Anand and VS Mani. Practices to make agile test teams effective: challenges and solutions. In 2015 IEEE 10th International Conference on Global Software EngineeringWorkshops, pages 7–11. IEEE, 2015.

[72] Rajeev Kumar Gupta, Prabhulinga Manikreddy, and KC Arya. Pragmatic scrum transformation: Challenges, practices & impacts during the journey a case study in a multi-location legacy software product development team. InProceedings of the 10th Innovations in Software Engineering Conference, pages 147–156, 2017.

[73] Laurie Williams. Agile software development methodologies and practices. In Advancesin Computers, volume 80, pages 1–44. Elsevier, 2010.

[74] Granville G Miller. The characteristics of agile software processes. In Technology of Object-Oriented Languages, International Conference on, pages 0385–0385, 2001.

[75] Pernille Lous, Paolo Tell, Christian Bo Michelsen, Yvonne Dittrich, and Allan Ebdrup. From scrum to agile: a journey to tackle the challenges of distributed development in an agile team. In Proceedings of the 2018 International Conference on Software and System Process, pages 11–20, 2018.

[76] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham,Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. Manifesto for agile software development. 2001.

[77] Rashmi Phalnikar, VS Deshpande, and SD Joshi. Applying agile principles for distributed software development. In 2009 International Conference on Advanced Computer Control, pages 535 – 539. IEEE, 2009.

[78] John Erickson, Kalle Lyytinen, and Keng Siau. Agile modeling, agile software development, and extreme programming: the state of research. Journal of Database Management(JDM), 16(4):88–100, 2005.

[79] PÄRJ ÅGERFALK and Brian Fitzgerald. Old petunias in new bowls? Communications of the ACM, 49(10):27, 2006.

[80] Maria Paasivaara and Casper Lassenius. Could global software development benefit from agile methods? In2006 IEEE International Conference on Global Software Engineering(ICGSE'06), pages 109–113. IEEE, 2006.

[81] Alistair Cockburn and Jim Highsmith. Agile software development, the people factor. Computer, 34(11):131–133, 2001.

[82] Erran Carmel and Ritu Agarwal. Tactical approaches for alleviating distance in global software development. IEEE software, 18(2):22–29, 2001.

[83] Usman Farooq and Muhammad Umar Farooq. Exploring the benefits and challenges of applying agile methods in offshore development, 2011.

[84] Sunila Modi, Pamela Abbott, and Steve Counsell. Exploring communication challenges

associated with agile practices in a globally distributed environment. InProc. of RAISE 2012-Researching Agile Development of Information SystEms Conference, 2012.

[85] Philipp Diebold, Constanza Lampasona, and Davide Taibi. Moonlighting scrum: An agile method for distributed teams with part-time developers working during non-overlapping hours. Eighth International Conference on Software Engineering and Advances, IARIA, pages 318–323, 2013.

[86] Hubert Smits and Guy Pshigoda. Implementing scrum in a distributed software development organization. InAgile 2007 (AGILE 2007), pages 371–375. IEEE, 2007.

[87] R Sriram and SK Mathew. Global software development using agile methodologies: A Review of literature. In 2012 IEEE International Conference on Management of Innovation & Technology (ICMIT), pages 389–393. IEEE, 2012.

[88] Barbara Kitchenham. Procedures for performing systematic reviews. Keele, UK, KeeleUniversity, 33(2004):1–26, 2004.

[89] Sally Rumsey. How to find information: a guide for researchers. McGraw-Hill Education(UK), 2008.

[90] Jane Webster and Richard T Watson. Analyzing the past to prepare for the future: Writing a literature review. MIS quarterly, pages xiii–xxiii, 2002.

[91] Siffat Ullah Khan and Mahmood Niazi. Critical challenges in offshore software development outsourcing: an empirical study. InInt. IASTED Conf. on Software Engineering SE, Greece, 2012.

[92] Staffs Keele et al. Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.

[93] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and AndersWesslén. Experimentation in software engineering. Springer Science & Business Media, 2012.

[94] Martin Ivarsson and Tony Gorschek. A method for evaluating rigor and industrial relevance of technology evaluations. Empirical Software Engineering, 16(3):365–395, 2011.

[95] Hussan Munir, Misagh Moayyed, and Kai Petersen. Considering rigor and relevance when evaluating test driven development: A systematic review. Information and Software Technology, 56(4):375–394, 2014.

[96] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. InProceedings of the 18th international conference on evaluation and assessment in software engineering, pages 1–10, 2014.

[97] Nicholas Clifford, Meghan Cope, Thomas Gillespie, and Shaun French. Key methods in geography. Sage, 2016.

[98] Jeremy Miles and Paul Gilbert. A handbook of research methods for clinical and health psychology. Oxford University Press on Demand, 2005.

[99] Eric Drever. Using Semi-Structured Interviews in Small-Scale Research. A Teacher ' sGuide.ERIC, 1995.

[100] Robyn Carston. Herbert h. clark, using language. cambridge: Cambridge university press, 1996. pp. xi+ 432.Journal of Linguistics, 35(1):167–222, 1999.

[101] Faiza Anwer, Shabib Aftab, SS Muhammad Shah, and Usman Waheed. Comparative Analysis of two popular agile process models: Extreme programming and scrum.International Journal of Computer Science and Telecommunications, 8(2):1–7, 2017.

[102] Ken Schwaber. Agile project management with Scrum. Microsoft press, 2004

[103] Margherita Pagani.Encyclopedia of multimedia technology and networking, volume 3. IGI Global, 2008.

[104] Raymond Opdenakker. Advantages and disadvantages of four interview techniques in

qualitative research. In Forum qualitative sozialforschung/forum: Qualitative social research, volume 7, 2006.

[105] Wes Williams and Mike Stout. Colossal, scattered, and chaotic (planning with a large distributed team). In Agile 2008 Conference, pages 356–361. IEEE, 2008.

[106] Giovanni Asproni. Motivation, teamwork, and agile development. Agile Times, 4(1):8–15, 2004.

[107] Tore Dybå and Torgeir Dingsøyr. Agile project management: From self-managing teams to large-scale development. In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, volume 2, pages 945–946. IEEE, 2015.

[108] Rashad Yazdanifard, Tshepo Molamu, and Mohammed Garba Musa. Project management and project integration management in relationship with service marketing. In 2011 International Conference on Management and Service Science, pages 1–4. IEEE, 2011.

[109] Johanna Kirsilä, Magnus Hellström, and Kim Wikström. Integration as a project management concept: A study of the commissioning process in industrial deliveries.International Journal of Project Management, 25(7):714–721, 2007.

[110] Barry Boehm and Hoh In. Identifying quality-requirement conflicts. IEEE software,13(2):25–35, 1996.

[111] Xiaoqing Liu, Chandra Sekhar Veera, Yan Sun, Kunio Noguchi, and Yuji Kyoya. Priority assessment of software requirements from multiple perspectives. In Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004., pages 410–415. IEEE, 2004.

[112] Hubert F Hofmann and Franz Lehner. Requirements engineering as a success factor in software projects. IEEE software, (4):58–66, 2001.

[113] Barbara A Kitchenham, Shari Lawrence Pfleeger, Lesley M Pickard, Peter W Jones, David C. Hoaglin, Khaled El Emam, and Jarrett Rosenberg. Preliminary guidelines for empirical research in software engineering. IEEE Transactions on software engineering, 28(8):721–734, 2002.

[114] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering – a systematic literature review. Information and software technology, 51(1):7–15, 2009.

[115] Rick W Wright, Richard A Brand, Warren Dunn, and Kurt P Spindler. How to write a systematic review. Clinical Orthopaedics and Related Research®, 455:23–29, 2007.

[116] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. qualitative research in psychology. Qualitative Research in Psychology, 3(2):77–101, 2006

[117] R Akif and H Majeed. Issues and challenges in scrum implementation. InternationalJournal of Scientific & Engineering Research, 3(8):1–4, 2012.

[118] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering.Empirical software engineering, 14(2):131, 2009.

[119] Alison Nightingale. A guide to systematic literature reviews.Surgery (Oxford), 27(9):381–384, 2009.

[120] Sebastian K Boell and Dubravka Cecez-Kecmanovic. A hermeneutic approach for conducting literature reviews and literature searches.Communications of the Association for Information Systems, 34(1):12, 2014.

[121] Lois E Reed. Performing a literature review. InFIE ' 98. 28th Annual Frontiers in Education Conference. Moving from Teacher-Centered Learner-Centered Education. Conference Proceedings (Cat. No. 98CH36214), volume 1, pages 380–383. IEEE, 1998.

[122] Robert K Yin.Case study research and applications: Design and methods. Sage publications, 2017.

[123] Elizabeth Bjarnason, Kari Smolander, Emelie Engström, and Per Runeson. A theory of distances in software engineering. Information and Software Technology, 70:204 219,2016.

APPENDIX

More information on SLR, Case Study can be found in the links below:

Link of articles used in SLR:
https://docs.google.com/spreadsheets/d/1cnCbnOEpPILE3T0Kl7ZJG2k5Q0Yq_tdFHkYw020StKI/edit?usp=sharing

Link of quality assessment results:
https://docs.google.com/spreadsheets/d/1gLx4OV98tAkpID23s3vSNTFLZurGXNltOclPc41ZP4/edit?usp=sharing

Link of original answers of interview:
https://docs.google.com/document/d/1LurFGgmR01tgFNJNVLdv9h8848pySVWK1WXz5kLvObw/edit?usp=sharing

Link of Code table:
https://docs.google.com/document/d/1UGbU0HTuSAhl9mnFsMXUqay5deZsGjaEm2bKsMjyOFo/edit?usp=sharing