Master of Science in Software Engineering
January 2020

A study of factors affecting requirement communication between different roles involved in requirements engineering process

Di Jiang
Chenxi Zhao

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):
Di Jiang
E-mail: diji17@student.bth.se

Chenxi Zhao
E-mail: chza17@student.bth.se

University advisor:
Dr. Krzysztof Wnuk
Department of Software Engineering

Faculty of Computing
Blekinge Institute of Technology
SE–37179 Karlskrona, Sweden

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

Abstract

Background. Requirements communication is an important activity that involves various stages in the project development process. Correctly understanding the requirements of stakeholders and effectively communicating them are key factors for the success of the project. In addition, it is critical to communicate any requirements changes during project development, so effective requirement communication plays an active role in project life cycle. In the existing research, many authors have reported the importance of requirement communication, but there are still research gaps in specific areas. Therefore, we focus our attention on requirement types, requirement volatility and participations' roles, and explore the impact of these aspects on requirement communication.

Objectives. In this thesis, we focused on the factors affecting requirement communication. The main objective is to understand and analyze the impact of requirement types and requirement volatility on requirement communication between different roles, and explore the challenges encountered in requirement communication. For the challenges that have been identified, we have summarized some solutions to achieve effective requirements communication.

Methods. In this thesis, we use systematic literature review (SLR) and survey methods. Firstly, the snowballing iteration method was used to perform five forward and backward snowballing iteration. 27 articles were identified to understand the current research process of requirement communication and to extract the challenges faced by requirement communication. Then conducted an online survey based on the results of the SLR, with the purpose of verifying the data obtained from the SLR and understanding the actual requirement communication status. Perform statistical analysis and narrative analysis on all results to identify the relationship between the data needed for our research.

Results. 27 articles are finally selected by us through SLR. Through the full context of those 27 articles, we identified 7 articles that related to different roles involved in requirement communication, 4 articles that mentioned requirement volatility, 11 articles that identified challenges encountered in requirement communication and 17 articles that provided solutions to improving requirement communication. Based on the answers from Survey, we identified the factors that affecting requirement communication from three aspects, which are requirement types, requirement volatility and different roles. What's more, a new challenge is identified from the answers of open-questions.

Conclusions. According to the analysis and results of SLR and Survey, we can clearly conclude that requirement volatility and requirement types will affect the selection of communication method and requirement volatility will further influence the final result of requirement communication. However, there is no significantly relationship between different roles and the selection of communication method. In addition, 8 challenges are identified from the analysis of SLR while a new challenge is identified from the answers of open-questions. Three types of solutions are also identified and organized from SLR, which are using models, patterns, frameworks, and other techniques, suitable consideration of communication methods and some specific suggestions.

Keywords: Requirement engineering, Requirement communication, Software development

Acknowledgments

First of all, we sincerely thank our supervisor Dr. Krzysztof Wnuk for his help. He provided us with a lot of professional resources and detailed modification suggestions. His conscientiousness helped us improve the quality of our research, and his supervision and encouragement supported us to complete the paper. He has provided the most important assistance to all of our work, and we are honored to work with him.

Then, we are very grateful to those who have helped us throughout the study and to the participants in the questionnaire. Their experience and answers helped us overcome many research difficulties, without their help we could not successfully complete the survey.

Finally, we would like to express our heartfelt thanks to our family and friends. It was the love they gave us behind the scenes that kept us going.

Contents

1. Introduction

Requirement communication can help stakeholders identify requirements at different stages of software development and understand different requirement changes. Identifying and understanding product requirements and negotiating requirements effectively can significantly increase the possibility of project success. Improper requirements communication can be one of the main causes of failure or delay in software project development, which will finally result in failing meet customers' expectations. In addition, stakeholders in different roles should also coordinate and communicate in different ways to support effective requirement communication [1]. Therefore, high effectively requirement communication can solve some problems that could lead to project failure [2].

In this case, we focus on how requirement communication is conducted in three different aspects, which are different roles of stakeholders, different types of requirements, and requirement volatility. By understanding whether requirement communication is affected by these three different aspects, we firstly conducted systematic literature review (SLR). We started from identifying the publication trend of articles related to requirement communication to summarize the study of previous articles related to the topic of our paper through systematic literature review, and we can further analyze research content of each article to get more meaningful results, such as summarized the challenges of requirement communication that had been identified in previous articles. Next, we conducted a survey through publishing an online questionnaire to identify the different communication methods used for different roles, and to understand which communication methods the respondents prefer to use among different types of requirement changes and the effectiveness of these communication methods. In addition, our survey can further verify the results of the SLR, which means using the results of the questionnaire to verify the frequency of challenges that have appeared in the articles and summarize some new challenges based on the answers of respondents.

The thesis is structured as follows. Chapter 2 introduces the research background and related work. Chapter 3 introduces the research aims, questions and research methods we chose, the motivation for choosing them and the research plan. Chapter 4 focuses on the results of SLR. It also shows the author's analysis of SLR results. Chapter 5 presents the data results of survey and analysis. Chapter 6 discusses the research questions and gives the threats to verify. Chapter 7 gives the conclusions and what needs to be done in the future.

2. Background

As Plachkinova et al. [2] said, “an effective requirement acquisition method play a key role in ensuring the rationality, correctness, and usefulness of the best requirements solution.” The final product requirements are agreed upon by different stakeholders through communicating and discussing different detailed requirements. Therefore, communicating and identifying these requirements plays a vital role in follow-up activities.

Requirement communication is an important activity in project development to understand the requirements of project stakeholders at different stages of the project life cycle and to inform stakeholders of any changes in requirements (final to solution scope) during project development. Understanding stakeholder requirements and how to realize them can greatly increase the likelihood of a successful project. Thus, requirement communication plays an active role in project development throughout the lifecycle.

In order for stakeholders to have a consistent understanding of the requirements of specific products, requirement communication activities between different stakeholders for different types of requirements and types of requirement changes are really important [3].

In most cases, software deployment cannot run smoothly due to misunderstanding of requirements during the requirements communication process. For example, inaccurate requirement communication can greatly contribute to failing to meet customers’ expectations and the cognitive distance among stakeholders. What’s more, the content of requirement communication will also involve some specifications of requirements, which are the definition of requirement specifications through existing domain knowledge about many social, administrative, functional and technical elements. Schneider et al. [4] pointed out that a software development project may involve many stakeholders with different roles, and whether the project can be successfully completed depends to a large extent on whether the stakeholders can effectively conduct requirement communication. Therefore, the challenges of requirement communication can be largely overcome through different requirement communication methods, and finally effective requirement communication can be achieved.

2.1 Related work

Nurmuliani et al. [5] investigated requirement volatility in the software development life cycle and emphasized that "communication and coordination among project members is very important because it allows them to track changes between affected products" [5], which means that the author believes in the importance of requirements communication when requirements changes occur, but the factors that may affect requirement communication are not analyzed comprehensively when requirements are changed. Therefore, we will study how to deal with different types of requirement changes through requirement communication and the different communication methods that can deal with different types of requirement changes.

Gorschek et al. [6] lists some knowledge about different requirement types, and Muhammad [51] also illustrates several of them. But they didn’t mention how to handle requirement communication regarding different types of requirements. Then we can research the impact of diversity of requirements types on requirement communication and the two articles might can give us some

knowledge or inspirits of requirement types. And Therefore, we will research different communication methods for different types of requirements.

Bjarnasson et al. [7] argues that "product complexity (RC2a) requires many skills scattered across many different roles", and the different roles of stakeholders may choose different communication methods for requirement communication. Although the nature of requirement communication requires at least two people to discuss, and the two people might play the same role. However, we want to focus on the different roles that involved in requirement communication process. But there are too many specific roles identified in different software companies, thus we decided to provide general roles that everyone knows and is familiar with, which can eliminate the ambiguity of roles recognition for peoples.

Cruzes et al. [40] stated how to implement communication between developers and testers while Abelein and Paech [34] studied how to improve communication between users and developers. Although most of authors study requirement communication between two specific roles, of which research direction is much more common nowadays, the research on the communication between two specific roles can also cause limitations in this field. Thus, we want to explore the impact on requirement communication not only limited within two specific roles, so we focus on more roles that involved in requirement communication. We will research to determine the responsibilities of stakeholders in different roles that are related to requirement communication activities and to further understand which communication methods stakeholders will use for their different responsibilities that related to requirement communication activities.

In addition, we also read a lot of other articles that related to requirement communication, such as Curtis et al. [8] pointed out in his research that practical requirement engineering has been observed in three dimensions of requirements engineering. Generally, requirements engineering starts with a detailed requirements document for the working system as a composition part of the entire software development process. Therefore, through effective communication, stakeholders will be agreed on one understanding of the requirements. In addition, there will be sufficient stakeholder protocol as well as well-designed rules and formats against specific requirements.

The same authors of [7] also pointed out that there are various communication gaps in the process of requirement communication. Through using semi-structured interview method, he identified four main factors that led to communication gaps, including scale, time, common views and decision-making structures. They also found that it may not be possible to achieve the ultimate customer expectations for the product under these communication gaps. Therefore, reducing the communication gap can help achieve effective requirement communication to ensure meeting customer expectations and further reduce software quality issues and energy waste.

In the software development life cycle, requirement communication is a intensive task, and geographical and time factors may affect the effectiveness of communication among stakeholders in distributed development. Damian et al. [9] evaluated the impact of synchronous text communications, such as computer-mediated communication (CMC) and face-to-face (F2F). In addition, in this empirical study proposed by Calefato et al. [10], they investigated the effects of requirement communication based on the communication model of asynchronous text and synchronous video conference according to the six teams participating in the negotiation process. The results show that

"requirement negotiations were more effective when the groups conducted asynchronous structured discussions of priority issues to the synchronous negotiation meeting." Although the two research literatures discuss the effectiveness of communication based on different requirement communication models and some challenges in requirements communication, they did not mention whether these two communication models will affect the effectiveness of communication according to the three aspects mentioned above called different roles, different types of requirements and different types of requirements changes, and the number of requirement communication models they studied is limited.

Schneider et al. [11] proposed that it is not sufficient to communicate requirements among stakeholders through using only written text formulation, other communication methods rather than documentation should be considered. Therefore, they proposed a method for visually acquiring requirements, including two types of symbols, which are document-based symbols for informal communication and the symbols for formal communication, and the difference between the two symbols for requirement communication has been evaluated in the latter study. In addition, they asserted that requirement communication is divided into solid and fluid information flows also remind us that communication methods can be divided into verbal and non-verbal forms, of which non-verbal communication form includes text and document communication types, while verbal communication form includes video conferencing, and teleconferencing types of informal communication.

After reviewing these literatures, we find that although there are many empirical studies related to requirement communication, there is no research investigating the factors that affect requirement communication comprehensively from the three aspects we mentioned previously. In this case, we decided to use SLR (Systematic Literature Review) and survey to investigate the factors affecting requirement communication and summarize the challenges that will be involved in the process of requirement communication.

3. Research Aims and objectives, Research Questions and Research Methods

3.1 Aim and objectives

The aim of this thesis is to analyze factors affecting requirement communication. In addition to that, we would like to propose the challenges that encountered in requirement communication and generate some improvements. We divide this aim into five objectives as follow

- O1: To identify the publication trends in requirement engineering articles regarding requirement communication.
- O2: To identify which communication methods are most commonly used in the requirement communication process between different common roles.
- O3: To identify whether various requirements types impact the requirement communication.
- O4: To analyze and identify the impact of requirement volatility on requirement communication.
- O5: To identify the challenges of requirement communication and summarize views for effective requirement communication.

3.2 Research Questions

In order to achieve the objectives mentioned above, we proposed 5 research questions which are shown as following:

RQ1: What are the publication trends in requirement engineering articles regarding requirement communication?

We hope to get the publication trends of articles related to requirement communication based on research RQ1, to determine whether it is necessary to summarize the previous articles related to the topic of our paper through systematic literature review, and we can further analyze research content of each article to get more meaningful results. We will record the publication year of each research article and perform a rigorous correlation score, and analyze the final data to study this question. Obtain the publication trends in requirement engineering articles regarding requirement communication by performing literature review to answer RQ1.

Expect outcomes: Sorting and grouping the publication years of related literatures in the literature review, and the result shows that the number of articles related to requirement communication is increasing.

RQ2: How different roles in requirements engineering communicate about requirements?

RQ2.1: *What are the responsibilities of the different roles, and which of them relate to requirements communication?*

RQ2.2: *What are the communication methods used by different roles?*

We decided to perform a survey, first determine the roles of the respondents in software engineering, the common roles of respondents involved in requirement communication will be identified from literatures, then we will identify the typical responsibilities of different roles through the online questionnaire, and then filter the responsibilities that related to requirements communication in order to get the answer of RQ2.1. The multiple choices questions will be set in the questionnaire to ask the respondents what communication methods they have used, and evaluate the effectiveness of each communication method to get the answer of RQ2.2.

Expected outcomes: For example, one of the roles of the respondents is project manager. We can find out the responsibilities of project manager through an online questionnaire, and then filter the responsibilities that related to requirements communication, such as market research conduction and requirements mining. The most effective communication method for the respondents will be selected by using matrix multiple choice questions. Based on this survey, we can know the corresponding most effective communication methods for different roles.

RQ3: Does various types of requirements impact requirements communication?

The type of requirement is a key indicator in the research of requirements, and we will take a survey to verify whether requirements at different types affect effective communication. Respondents in the online questionnaire play different roles in requirement communication, and they are asked to choose the most common type of requirement they encountered in their requirement communication experience, and whether they will adopt different communication methods for different types of requirement. After summarizing the answers of open-questions and close-questions from the survey, it can be judged whether different types of requirements will affect requirements communication. Then RQ3 will be answered through the survey.

Expect outcomes: For example, among all respondents, the most common type of requirements is functional requirement, but most respondents will change the communication method based on the different types of requirements.

RQ4: What is the impact of requirement volatility on requirement communication?

RQ4.1: *What is the impact of different kinds of requirement changes on requirement communication?*

We first determine the types of requirement changes, and then conduct a survey to get the most types of requirement changes the respondents propose or obtain. The answers of the questionnaires are used to identify the communication methods used by end users to communicate requirements changes to the software development team, and how software engineering practitioners communicate requirement changes within the team. In addition, we can know whether the requirement changes have been successfully implemented according to the answers of questionnaire. Through the analysis of the answers of these questions, understand whether the types of requirement changes will affect requirement communication, then the answer of RQ4.1 is obtained.

Expected outcomes: For example, in the questionnaire for end users, "Additional Function" is the type of requirement change that most frequently raised. They chose face-to-face communication method to communicate that type of requirement change to the development team, and track whether this requirement change was successfully implemented within a certain period of time. In the questionnaire for software engineering practitioners, for example, "Field Modification" is the type of requirement change that most frequently raised. Most of them choose the requirement documentation method to conduct communication, because the communication method in such form of text can clearly show the

written information to the end user. After collecting the answers data from respondents, most requirements changes were successfully implemented. Therefore, we can determine which aspects of requirement communication will be affected by requirement volatility.

RQ5: What are the challenges encountered in requirement communication and how to solve them?

We summarized the challenges of requirement communication that identified in previous papers through systematic literature review, and then verify the frequency of these challenges through survey, then propose some new challenges from respondents. In addition, we will summarize the results of literature review to propose some methods that can be used to optimize requirement communication. The results of literature review and survey will be combined together to answer RQ5.

Expect outcomes: For example, through the systematic literature review, we identified 5 challenges of requirement communication, among which challenge 1 and challenge 3 were the most common in the actual requirements communication process. And we found a new challenge through the survey results. We also got three methods to improve requirements communication from the literatures.

3.3 Systematic literature review research method

A systematic literature review can "address one or more research questions by identifying, rigorously evaluating, and integrating the results of all high-quality individual studies." [12] We hope to achieve most or all of the following goals by implementing a good systematic literature review [13] [14] [54]:

- Determine the extent to which existing research on requirement communication has progressed;
- Learn about the impact of different roles, requirement types or requirement volatility on requirement communication, and discuss how to improve requirement communication;
- Propose useful points for our research by understanding the topics of literatures [15];
- Comment, evaluate, and present arguments of the literatures for further researches;
- Determine the direction of our future research.

"A literature review is the basis for research in almost every academic area. This research process includes an understanding of current knowledge, substantive findings, and theoretical and methodological contributions to specific topics. Literature reviews are secondary sources that do not report new or original research work." [12] There are several different ways to conduct systematic literature review, including snowballing and database search methods. In the motivation part we will explain the reasons for choosing snowballing method.

By studying the existing literatures, we plan to summarize the challenges encountered in the process of requirement communication and the solutions to improve requirement communication in these literatures, and initially find out the factors that affect requirement communication but have not been studied specifically in the literatures. This can also provide us some more effective recommendations.

3.3.1 The motivation that using snowballing iteration method for SLR

Based on the System literature review (SLR) method proposed by Baumeister et al. [13], we learned that there are two main methods for implementing SLR, i.e. snowballing and database search methods. The database search method is based on a well-defined search string, and starting to find relevant literatures by systematic search in the databases. Although plenty of literatures about the keywords set

at the beginning is searched out, the study contents of most articles may deviate from our research field, and a large number of irrelevant articles may appear by using database search [14]. However, snowballing method will use the literatures with related topics in the related research field as a start set, and then use the articles in the start set to promote next iteration. According to Wohlin's guidance [1], we understand that snowballing is easier to implement than database search since after reading the title and abstract of the article, we can easily identify the articles related to our research topic and it won't be such easy for database search to identify related articles. For example, database search will be far less effective than snowballing when we define the key strings that are too general (e.g. "communication"). Wohlin [1] also suggested that "snowballing is particularly useful for extending systematic literature studies, since new studies almost certainly must cite at least one paper among the previously relevant studies or the systematic study already conducted in the area. Therefore, snowballing is a better method that can be used to extend systematic literature studies than database search. "

According to the conducting method of SLR proposed by Watson et al. [16], we will systematically organize and review the literatures by forward snowballing (i.e. the citation part of one literature) and backward snowballing (i.e. the reference part of one literature). Hence, we extract the basic steps of the snowballing method next based on the guidance of Wohlin [1], and the specific implementation steps are as Section 4.1.

3.3.2 Snowballing Procedure

This process involves two steps, including identifying the start set and then performing the forward and backward iterations based on the start set papers.

The diagram of the explanation of snowballing procedure is shown as Figure 3.1:

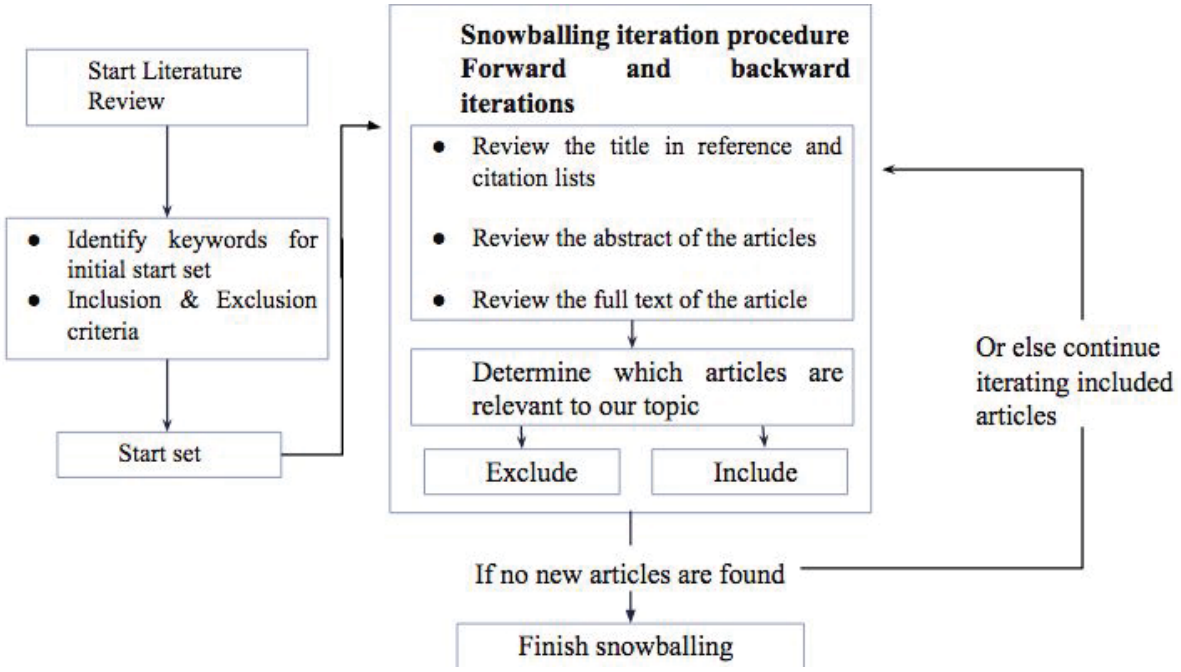

Figure 3.1: The flowchart of snowballing procedure

Identifying the start set papers. The first step is finding more relevant papers that apply on the backwards and the forward iterations.
Wolhin [1] observes on the following characteristics.

- Ensure the papers are obtained from different authors, publishers, years and communities.
- Starting size depends on the scope area that the study considers.
- Selections of the papers that are relevant in the case of the more articles are found.
- Keywords are used in framing the string along with their synonyms.

Based on the steps mentioned above, we first identified some keywords related to our topic, which are shown as follow:

K1- Software project management
Extend Keywords: Software development, Roles in software development team, Software development process

K2- Requirement engineering
Extend Keywords: Requirement management, Requirement types, Requirement volatility, Requirement changes, Requirement elicitation, Market-driven requirement management

K3- Requirement communication
Extend Keywords: Communication artifacts, Communication modes, Communication gaps, Communication problems, Requirement coordination, Requirement negotiation

Then we use the keywords to get search results in the database. After searching the keywords, we first received 3367 articles as an initial start set. In addition, our supervisor has helped us provide some literature related to our research field. Finally, 11 articles has been selected as a start set for snowballing. The procedure of start set identification is shown as Figure 3.2.

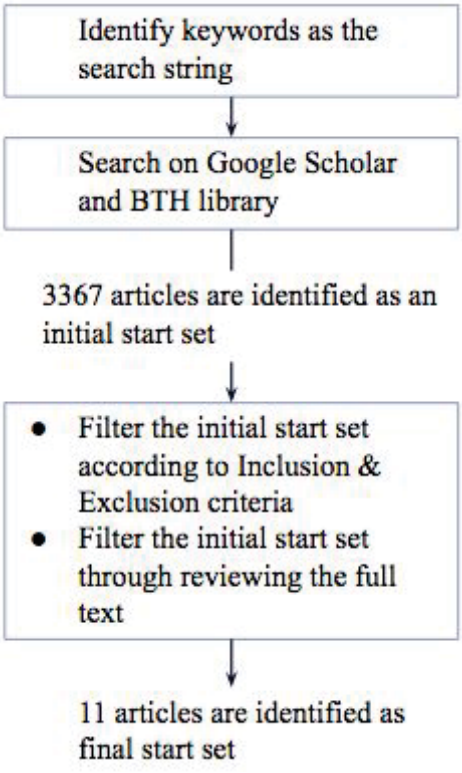

Figure 3.2: The flowchart of start set of identification

3.3.2.1 Inclusion and Exclusion criteria

The inclusion and exclusion criteria that we defined for selecting related literatures are as follow:

Inclusion criteria:
IC1: Literatures that are available in full text.
IC2: Literatures that are related to the field of requirement engineering.
IC3: Literatures that are mainly talking about requirement communication.
IC4: Literatures that are related to different types of requirements, such as use case requirements, functional requirements, quality requirements and so on.
IC5: Literatures that are related to the requirement communication with different roles or different teams.
IC6: Literatures that talked about requirement volatility or knowledge about requirement changes.
IC7: Literatures that are mainly to talk about the solution for improving requirement communication.
IC8: Literatures that are available after 2000.

Exclusion criteria:
EC1: Abstract is not available to be read.
EC2: Literatures that are mainly talking about communication patterns rather than the differences among those communication patterns.
EC3: Literatures that are mainly talking about the importance for requirement communication rather than how to improve it.
EC4: Literatures that are mainly talking about the generately challenges and difficulties for requirement communication.

3.3.2.2 Forward and backward snowballing iterations

The forward and backward snowballing iterations is the main step of our snowballing process. We have done 5 forward iterations and 5 backward iterations in total. The results of each review are determined by us and our supervisor, and the articles we consider appropriately will be used for the next iteration. The articles are taken by following the inclusion and exclusion criteria we wrote above. The backward snowballing is based on screening the reference of each paper selected. Forward snowballing is performed by going through the citations of the paper. Usually, we review the articles based on what is shown in the Google Scholar. In the end, when we find out that there is no articles related to our topic, we ended the entire literature review process.

3.3.3 Quality assessment

We evaluated the quality of each article according to the guidelines and checklist of Rigor and Relevance by Gorschek et al. [46]. Relevance score can clearly show the degree of relevance of specific article to the current research field while rigor can clearly specify the limitations of the research conducted by the article and the feasibility of the research method, thereby the reliability of the research results of specific articles can be determined eventually.

The relevance/rigor of each article is carefully assessed and graded by us, and the final scoring results are also reviewed by our supervisor, then the quality of each article can be clearly assessed by the

scores of rigor/relevance. The detailed tables and charts for rigor/relevance scores are shown in Appendix A and Figure 4.3.

3.3.4 Data extraction

In order to realize the maximum value of literature review and get as much effective information as possible from each literature, it is very important to extract reasonable data. Based on our research questions and objectives, we determined the content to be extracted from each article and developed a table. Data extraction tables can help us extract useful information at any time when we read them, and also facilitate data review when needed. For the formulation of the data extraction table, we refer to the guidance of Keele [56] and combine it with our actual needs. As shown in Table 3.1, we mainly extracted the information from the article.

- Basic information of the article: In the literature review screening process, the first step of our review is to preliminarily determine whether the article is relevant to our research direction based on the title, abstract and conclusion. In addition, it is useful to avoid duplication. We also recorded the publication year of the article to exclude articles that did not meet the criteria we formulated, and conducted a publication trend analysis.
- Study type and research method: According to Wierlinga's method [45], we will identify the study type of the article from the following aspects: evaluation research, proposal of solution, validation research, philosophical papers, opinion papers, personal experience papers. The research methods used in this article can help us better understand the specific work and opinion extraction performed by the author.
- Study context: after reading the abstract and conclusion of the article, we need to further understand the specific content of the article. Only by understanding the full text we can extract as many questions related to our research direction as possible, which is conducive to answer specific research questions.

Table 3.1：Data extraction form

Data item		**Value**	**Related Questions**
Basic Information	Paper ID		RQ1, RQ2, RQ3, RQ4, RQ5
	Title		
	Publication year		
	Abstract		
Study types			RQ1
Research methods			RQ1
Study Content	Does the article mention different roles in requirement communication?		RQ2
	Does the article mention different types of requirements?		RQ3

	Does the article mention requirement volatility or requirement changes?		RQ4
	Does the article mention the challenges of demand communication		RQ5
	Does the article suggest a solution to the problem of requirement communication?		RQ5

3.4 Survey research method

3.4.1 Online questionnaire

Since our research is based on the current requirement communication challenges encountered during the development process, and our research is mainly related to the software engineering practitioners and end-users, it is necessary to regard them as respondents to our research. Because of the different work experience and opinions of respondents, a lot of practical data from their answers can be analyzed and answered to our research questions.

We decided to select the survey method and conduct it as the form of questionnaires. The first advantage of questionnaire is that it can save much time and human resources while get more opinions and perspectives on requirement communication from the answers of respondents who have different job positions and work experience. Second, because questionnaire is a kind of structured method, the sequence of questions presented by the form of words and the alternatives of answers provided for each question will not be changed, thus, it is easy to quantify the survey results. Third, the results of the survey method are easier to statistic and analyze. What's more, survey method can be conducted in a large-scale. We can understand the opinions, thoughts, attitudes, and behaviors of the respondents from the survey method.

In order to getting a plenty of data in a short time period, we chose to publish our questionnaires online. Online questionnaires not only allow us to reach more respondents from different geographical locations [17], but also to filter and analyze data. According to the two types of respondents related to our study participants i.e. software engineering practitioners and end-users, the questionnaires will be divided into two different types.

3.4.2 The reason for not considering other research methods

We first considered various other research methods, such as case study, experiment, interview and so on. For case study, it is used to research specific systems and to generate different specific cases in order to analyzing data. However, the data that collected through case study may not be comprehensive, and our research should focus on software engineering practitioners and end-users based on different environmental factors rather than specific environmental factors or specific cases. Therefore, case studies should not be considered. As for interview, it will cost much on time while we ask specific questions to specific respondents and the data collected may be insufficient due to the

population of our interviewees is not large enough, so interview method is excluded. As for experiment, because it is necessary to take into account the impact of different factors and variables, it is not appropriate to use in large-scale research field and can only be achieved in a small-scale specific environment, and experiment is mainly to explore the causal relationship between a small number of variables and the effect of communication, the research object is usually a "small sample". However, our research requires extensive observations or interviews with a large number of respondents. What's more, the research environment for experiment is in a laboratory artificial environment but our research will be conducted in a wide social environment.

3.4.3 Survey Planning and Designing

We divide the survey into three processes: survey planning, survey design and survey execution, and divide each process into three specific steps, which are detailed in Figure 3.3.

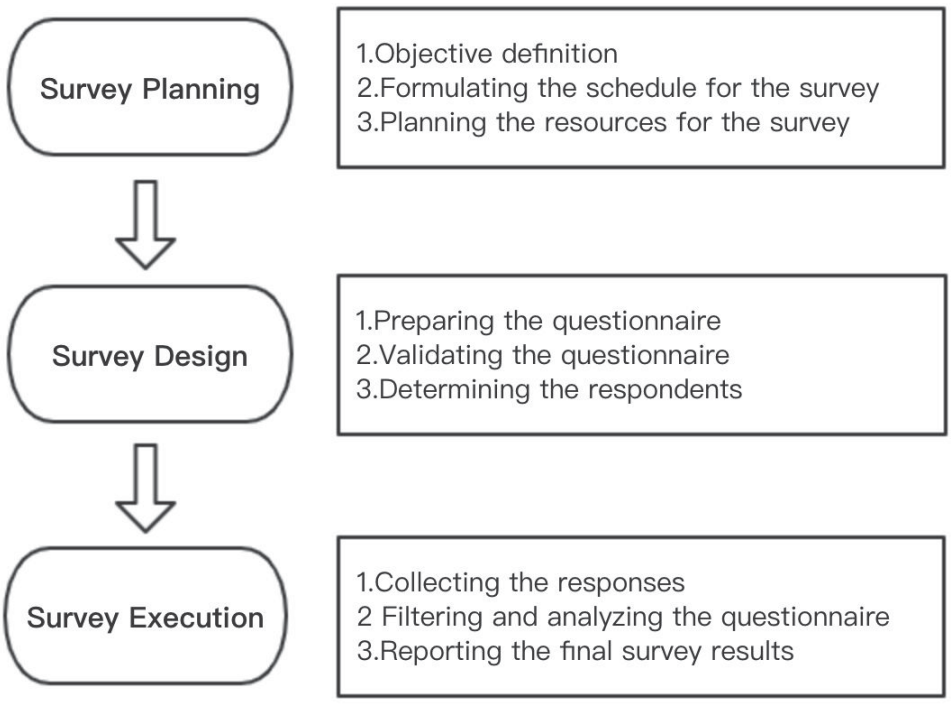

Figure 3.3: The survey planning procedure

3.4.3.1 Survey Planning

1). Objective definition

Defining the survey objectives based on our research questions above is the first thing we have to do. Only after we have identified the survey objectives, can we identify other survey content, such as the respondents we want to select, the data we collect through surveys, and so on. Therefore, defining objectives is the primary task of planning a survey. In our study, the objectives of this survey is as follows:

- To identify the communication methods that participants use in the requirements management process
- To identify the challenges that participants face in the process of demand communication

2). Formulating the schedule for the survey
Considering the completion time of the literature review and the submission of the paper, we plan to conduct a six-week online survey from September 16, 2019 to October 27, 2019. The recycling of the questionnaire was strictly stopped at the deadline.

3). Planning the resources for the survey
Due to the limited range of people we have contacted, we have chosen an online survey that will expand the range of respondents and more easily collect and count the results of the questionnaire. We plan to send the survey to some common social networking platforms, including Facebook, CSDN, and the software that are widely used in China, i.e. WeChat, QQ. We also recommend that respondents forward the questionnaire link to friends in related fields to expand the scope of the release.

3.4.3.2 Survey Design

1). Preparing the questionnaire
We selected the online questionnaire to conduct a survey and use a questionnaire development software called wjx to design and share. For research objectives, in order to better obtain empirical data on both the software engineering practitioner and the end user, we decided to let the respondents of both identities answer different questions.

The questionnaire is divided into three parts, and the questions are divided into open and closed questions. Use closed questions to get the fixed information we have to know. The open question is based on the fact that we may not cover a wide range of closed questions.

The first part of the questionnaire has only one question, which is used to check the identity information of the participants, and is used to jump to the part corresponding to their identity to continue to answer questions.

The second part is a questionnaire for software engineering practitioners. All questions are based on their professional understanding and experience in requirement engineering and communication.The third part is the questionnaire for the end user, we focus on the end user's participation experience, and ask questions in a more concise language.

The closed question is to get the following basic information:

1.The role of respondents in software development projects
2. Respondent's team size, work experience and knowledge level
3. Requirements communication methods adopted by respondents in software development projects
4.The methods used to exchange information with software development teams or users during requirement communication
5. The type of requirement changes they have proposed or encountered
6. Requirement communication challenges faced by respondents

In addition, we also set a lot of open questions and “other” options. These questions are set to give respondents more opportunities to express their views.
At the end of the questionnaire, if the respondents are willing, we hope they can leave their email address or other contact information.

2). Validating the questionnaire
We believe it is necessary to conduct a validation of our questionnaire before the formal investigation to verify the feasibility of the survey and to exclude some irregularities of the questions.
We conducted a preliminary investigation of the questionnaire we designed with reference to the method proposed by Kitchenham and Pfleeger [18]. We invited 5 participants to participate in the experiment and record their answer time. Their average response time is 10 minutes, and we think it means the number of questions is reasonable. Based on their suggestions, we changed some incomprehensible problem descriptions and added some new ones. Finally, we wrote a formal questionnaire in English and Chinese. They can be viewed in the appendix.

3). Determining the respondents
Our planned respondents should have one of the following two characteristics:
- Software engineering practitioner with experience in software development team
- Software end users with experience in requirements communication with software engineering practitioner

We learn Kitchenham et al [18] divided sampling method into two categories, the improbability sampling and probability sampling. As a result of our social circle can meet the above characteristics of the person is less, so in order to gain more empirical data, we decided to adopt the snowball sampling of the improbability sampling.

We plan to continue to share the questionnaire with the respondents who meet the conditions of our respondents, and have been snowballing until the deadline for the questionnaire. In this way, we can investigate as much roles as possible in the field, such as requirement engineers, project managers, testers, developers, analysts, etc.

3.5 Survey and SLR data analysis

In general, data analysis is the process of classifying the original data and further transforming it into a series of data that can be used to organize and summarize research results [19]. The results of data analysis can be used to answer research questions and test research hypotheses to further draw conclusions for the research [19]. Data analysis can be divided into two kinds of methods, which are qualitative analysis and quantitative analysis and the two methods can be adopted according to the type of collected research data [20]. In our research, we plan to combine qualitative analysis with quantitative analysis methods to analyze the collected data. Because the qualitative analysis method is generally suitable for written language description, and the quantitative analysis method is based on statistical data to summarize the research results in mathematical language and according to the guidance of [21], we can understand that "the quantitative analysis method is mainly based on the intuition and experience of the researchers, and the collected data is used to analyze the nature, characteristics and changes of the research object, while the qualitative analysis method is mainly used to draw research results in non-data form based on the hypotheses and theories of the research. " On

the one hand, we use qualitative analysis method to analyze SLR data (Narrative analysis) by clarifying the topics discussed in each article and their relevance to our research questions, which is shown in Section 3.2. On the other hand, we will use statistical analysis to analyze the data of the questionnaires for quantitative analysis (Chi-square test, Fisher's exact test, Histogram).

3.5.1 Statistical analysis

Chi-square test is a kind of non-parametric test. Its basic principle is to calculate the degree of deviation between the actual observed value and the theoretical inferred value of the sample. If the chi-square value is larger, the degree of deviation between the two variable is larger; on the contrary, the deviation between the two variable is smaller. If the two values are completely equal, the chi-square value is 0, which indicates that the theoretical value is completely consistent. The specific steps are as follows.

1. First make a hypothesis, its null hypothesis H0 is: the observation frequency has no obvious relationship with the expected frequency. Then make an alternative hypothesis H1 which is opposite to the null hypothesis, that is, there is a relationship between the two.

2. Determine that the confidence is 95%, that is, the probability of rejecting H0 is 0.05.

3.Analyze the gap between the expected value and the observed value according to the formula, and calculate the X^2 and degree of freedom.

Chi-square's calculation formula is as follows [55]:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

O = the actual frequencies
E = the theoretical frequencies

The formula for calculating df:

df =(c-1)(r-1)

c: number of columns
r: number of rows

4. According to the chi-square and the degree of freedom, the P value can be obtained by looking up the chi-square distribution table, but in this study, we used the SPSS tool. If p-value<0.05, we will reject H0, which means the two variables are relevant; If p-value>0.05, then H0 is accepted, and H1 is not valid, which means the two variables are independent and irrelevant.

It is generally believed that for values in the chi-square test, the minimum expected frequency is required to be greater than 1, and at least 80% of cells expect frequency to be greater than 5, then the probability value calculated by using chi-square test is accurate. If the data does not meet the requirements, the Fisher's exact test then can be used to calculate the probability.

Fisher's exact test

When the data do not meet the requirements of chi-square test, that is, when the expected frequency of more than 20% of the cells is less than 5, we will adopt Fisher's exact test. The chi-square test relies on approximations, and Fisher's exact test is an exact test. Since the formula of fisher's exact test uses factorial and the data calculation is very accurate, we use SPSS tool to carry out.
The procedure of the Fisher's exact test is similar to that of chi-square test in that the p value obtained is used to judge the feasibility of H0.

Histogram

As the conception proposed by Karl Pearson [22], a histogram is a statistical chart and an estimation of the probability distribution of continuous variables which can represent the data accurately. After combining the data of different roles of the respondents with the data of different requirement types that collected from the questionnaire, we found that this is a two-dimensional data set, which means each data point includes two values (e.g. x and y), but there is only one dimension that needs to be compared to show (i.e. the number of respondents selecting specific different types such as requirement types, requirement change types and so on) in order to visually see the differences of data.

Histograms can not only visualize complex statistical data, but also make it clear for the difference of the column's heights, which let us to understand and further compare the corresponding data size easily. In addition, we also used pie chart to visually represent the distribution of respondents' roles, thus, the number of different respondents' roles participating in this survey is clearly to see in the next part.

3.5.2 Narrative analysis

We conducted a narrative analysis of the selected articles according to the guidelines of Rodgers et al. [23]. The authors said that the narrative analysis for data extraction of systematic literature review has two main steps, which are showing as follows:

1. Identification of relevant methodologies used in selected literatures [23]: In order to conduct this step, we carefully read the articles involved in each iteration and identified the research topics discussed in each article. We first roughly selected the articles by reading the title and abstract of each article, and then selected 27 articles related to our research area after reading the full text of each article. The 27 selected articles have also been reviewed strictly by our supervisor.

2. Creating guidance document [23]: The authors presented a guidance document for the classification of narrative analysis. It consists of four steps as below:
- Developing a Theory: We developed a theory based on the guidance of [24] and the aim and objectives of our research, the theory can not only help us identify which research questions and types of study in these articles need to be reviewed and interpret the final results of reviewing.
- Developing a Preliminary Synthesis: This step involves a preliminary integration and analysis of the selected articles. The guidelines suggested that "how a reviewer approaches the preliminary synthesis … will depend in part on whether the evidence to be synthesised is quantitative, qualitative or both." [24] Through this second step, we can summarize the final conclusions of each article to discover the challenges of requirement communication, and then further list the solutions that can solve these challenges and explore the potential relationship between the solutions and these challenges.

- Exploring relationships within and between studies: The third step involves investigating the relationships between the studies. According to one of the techniques mentioned in this guideline, we can get some information of relationships from the extracted data and their intervention work. We have conducted quality assessment for exploring the relationships among the quality of those articles, which is shown in Section 4.2.2.
- Assessing the Robustness of the Synthesis Product: The guidelines of [23] suggested that four criteria can be used to appraise the robustness of each article, which are "i. the study's methodological soundness, ii. the appropriateness of the study design to answering the review question, iii. the study relevance, and iv. an assessment of the overall weight of evidence which the study provides." [23] However, our comprehensive analysis of each article has been reviewed by our supervisor and the analysis of the article has been modified based on the advice of our supervisor. Then we can assess that the robustness of each article is relatively high eventually.

4. Results and Analysis of SLR

4.1 Results

We performed five snowball iterations, and finally selected 27 articles related to our research topics from total 1034 articles. The analysis of the publication trend of these articles, the types of articles, and the quality assessment of the content of articles are as follows.

4.1.1 Start set

Through the suggestions of our supervisor and the article reading when writing the proposal, we selected 11 articles as the start set articles. Of these 11 articles, 7 are conference articles and the remaining 4 are journal articles. The two databases used in the extracting of the applied data are Google scholar and BTH library. The Table 4.1 below defines the various articles selected for the study.

Table 4.1: Start set

ID	**Start set Articles**
P1 [3]	Al-Rawas, A., & Easterbrook, S. (1996). Communication problems in requirements engineering: a field study.
P2 [7]	Bjarnason, E., Wnuk, K., & Regnell, B. (2011, August). Requirements are slipping through the gaps—A case study on causes & effects of communication gaps in large-scale software development. In *2011 IEEE 19th international requirements engineering conference* (pp. 37-46). IEEE.
P3 [10]	Calefato, F., Damian, D., & Lanubile, F. (2007, August). An empirical investigation on text-based communication in distributed requirements workshops. In *International conference on global software engineering (ICGSE 2007)* (pp. 3-11). IEEE.
P4 [9]	Damian, D., Lanubile, F., & Mallardo, T. (2008). On the need for mixed media in distributed requirements negotiations. *IEEE Transactions on Software Engineering*, *34*(1), 116-132.
P5 [4]	Schneider, K., Stapel, K., & Knauss, E. (2008, September). Beyond documents: visualizing informal communication. In *2008 Requirements Engineering Visualization* (pp. 31-40). IEEE.
P6 [11]	Stapel, K., Knauss, E., Schneider, K., & Zazworka, N. (2011, August). Flow mapping: Planning and managing communication in distributed teams. In *2011 IEEE Sixth International Conference on Global Software Engineering* (pp. 190-199). IEEE.
P7 [25]	Coughlan, J., & Macredie, R. D. (2002). Effective communication in requirements elicitation: a comparison of methodologies. *Requirements Engineering*, *7*(2), 47-60.
P8 [2]	Plachkinova, M., Peffers, K., & Moody, G. (2015, May). Communication artifacts for requirements engineering. In *International Conference on Design Science Research in Information Systems* (pp. 104-118). Springer, Cham.

P9 [5]	Nurmuliani, N., Zowghi, D., & Powell, S. (2004, April). Analysis of requirements volatility during software development life cycle. In 2004 Australian Software Engineering Conference. Proceedings. (pp. 28-37). IEEE.
P10 [26]	Liskin, O. (2015, March). How artifacts support and impede requirements communication. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (pp. 132-147). Springer, Cham.
P11 [27]	Bjarnason, E., & Sharp, H. (2017). The role of distances in requirements communication: a case study. *Requirements Engineering*, *22*(1), 1-26.

4.1.2 Iteration 1

For the forward and backward snowballing of the first iteration, we found there are many articles related to our research topic. A total of 1034 articles were involved, and finally we selected 9 articles for the next iteration. The selection had to use on the inclusion and the exclusion criteria for selecting the most suitable articles. In this process, first of all, according to the title of the article, we chose 89 articles, and the remaining 945 articles were excluded because the topic has no relationship with our research field. Then we decided to exclude 61 articles after reading the abstract of each article and discussing with our supervisor. After reading the 28 articles left, 8 articles are duplication, and the context of 11 articles are not relevant to our research, so we finally chose 9. Of the selected articles, 8 are conference articles, and the remaining one is journal articles. The selected 9 articles are detailed in the Table 4.2.

Table 4.2: Results of iteration 1

ID	**Articles Relevant to our field**
P12 [28]	Fricker, S., & Glinz, M. (2010, September). Comparison of requirements hand-off, analysis, and negotiation: case study. In *2010 18th IEEE International Requirements Engineering Conference* (pp. 167-176). IEEE.
P13 [29]	Marczak, S., Damian, D., Stege, U., & Schröter, A. (2008, September). Information brokers in requirement-dependency social networks. In 2008 16th IEEE International Requirements Engineering Conference (pp. 53-62). IEEE.
P14 [30]	Niinimaki, T., Piri, A., Lassenius, C., & Paasivaara, M. (2010, August). Reflecting the choice and usage of communication tools in gsd projects with media synchronicity theory. In *2010 5th IEEE International Conference on Global Software Engineering* (pp. 3-12). IEEE.
P15 [31]	Stapel, K., Knauss, E., & Schneider, K. (2009, August). Using flow to improve communication of requirements in globally distributed software projects. In *2009 Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills* (pp. 5-14). IEEE.
P16 [32]	Niinimaki, T., Piri, A., & Lassenius, C. (2009, July). Factors affecting audio and text-based communication media choice in global software development projects. In *2009 Fourth IEEE International Conference on Global Software Engineering* (pp. 153-162). IEEE.

P17 [33]	Gallivan, M. J., & Keil, M. (2003). The user–developer communication process: a critical case study. *Information Systems Journal*, *13*(1), 37-68.
P18 [34]	"Abelein, U., & Paech, B. (2012, June). A proposal for enhancing user-developer communication in large IT projects. In *2012 5th International Workshop on Co-operative and Human Aspects of Software Engineering (CHASE)* (pp. 1-3). IEEE.
P19 [35]	Abelein, U., & Paech, B. (2014, April). State of practice of user-developer communication in large-scale it projects. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (pp. 95-111). Springer, Cham.
P20 [36]	Marczak, S., & Damian, D. (2011, August). How interaction between roles shapes the communication structure in requirements-driven collaboration. In *2011 IEEE 19th International Requirements Engineering Conference* (pp. 47-56). IEEE.

4.1.3 Iteration 2

For the second iteration, we referred to the total of 429 articles, there are 228 references while the rest of 201 articles are from citations. We reviewed those articles thoroughly based on the process that we followed for selecting related articles and no any other articles we could find that are related to our research topic. In this process, we selected 72 relevant articles based on their topics from the title screening, then several articles are selected after reading their abstracts from the articles of title selection. The final review observed what was approved by the professor. After removing 11 duplicate articles, there are 5 articles that we believe are the most relevant to our research area and can be used for the next iteration, which includes 2 conference articles and 3 journal articles. The following Table 4.3 shows the results of this second iteration.

Table 4.3: Results of iteration 2

ID	**Articles Relevant to our field**
P21 [37]	Fricker, S., Gorschek, T., Byman, C., & Schmidle, A. (2010). Handshaking with implementation proposals: Negotiating requirements understanding. *IEEE software*, *27*(2), 72-80.
P22 [38]	Lang, M., & Duggan, J. (2001). A tool to support collaborative software requirements management. *Requirements Engineering*, *6*(3), 161-172.
P23 [39]	Korkala, M., Abrahamsson, P., & Kyllonen, P. (2006, July). A case study on the impact of customer communication on defects in agile software development. In *AGILE 2006 (AGILE'06)* (pp. 11-pp). IEEE.
P24 [40]	Cruzes, D. S., Moe, N. B., & Dybå, T. (2016, August). Communication between developers and testers in distributed continuous agile testing. In 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE) (pp. 59-68). IEEE.
P25 [41]	Rodina, A., Amjed, T., & Zarinah, M. K. (2012, March). An empirical assessment of the use of different communication modes for requirement elicitation and negotiation using students as a subject. In *2012 IEEE Symposium on Computers & Informatics (ISCI)* (pp. 70-74). IEEE.

4.1.4 Iteration 3

Through the third snowballing iteration, there are 250 related articles, of which 116 are gotten from forward snowballing iteration and 134 are from backward snowballing iteration. In the end, only 1 conference article is related to our research and used for the next iteration. In the process, we first browsed their topic and remain 68 articles, 58 articles were excluded after reading the abstract and 3 were excluded due to the full text. After that, we excluded four duplicate articles. We found that two articles were published in 1996 and 1983 respectively, considering about the premature publication year, we decided to remove them from the iteration list. The final article selection results are decided through combining the opinions of us and our supervisor and are shown in the Table 4.4.

Table 4.4: Results of iteration 3

ID	**Articles Relevant to our field**
P26 [42]	Fricker, S., Gorschek, T., & Glinz, M. (2008, September). Goal-oriented requirements communication in new product development. In *2008 Second International Workshop on Software Product Management* (pp. 27-34). IEEE.

4.1.5 Iteration 4

In the fourth forward and backward snowball iteration, there are total of 49 related articles, of which 37 are from references and the remaining 12 are from citations. During this iteration, there are 16 articles excluded as the topics are not relevant to our research topic. After reviewing and discussing about the remaining 33 articles with our supervisor, 28 articles are removed based on the irrelevant abstract and 4 articles are removed due to duplication. In the end, we selected 1 conference article for the next iteration shows in Table 4.5.

Table 4.5: Results of iteration 4

ID	**Articles Relevant to our field**
P27 [43]	Kwan, I., Damian, D., & Storey, M. A. (2006, September). Visualizing a requirements-centred social network to maintain awareness within development teams. In *2006 First International Workshop on Requirements Engineering Visualization (REV'06-RE'06 Workshop)* (pp. 7-7). IEEE.

4.1.6 Iteration 5

In this fifth iteration, we reviewed 12 references and 12 citations of literature P27. And 4 articles are selected after reading the abstracts and discussing with our supervisor. However, we noticed that those 4 articles are all already involved in the previous iterations and examined thoroughly before. After discussing with our supervisor, we could not find any article to continue to the next iteration, which means there is no any other articles that related to our research area for the next forward and backward snowballing iteration. Therefore, we end up this iteration, and thus our snowballing iteration was finally done until this fifth iteration.

4.2 Distribution of relevant articles

After five snowballing literature iterations, we selected 27 articles related to our research topic. But because we limited many criteria, such as publication year, language, etc., there are three articles that were excluded because they were published before 2000. However, although P1 was published in 1996, we believe that it is related to our research direction and has relative high research value, so we decided to retain it. Figure 4.1 shows the distribution of selected articles based on the year of publication.

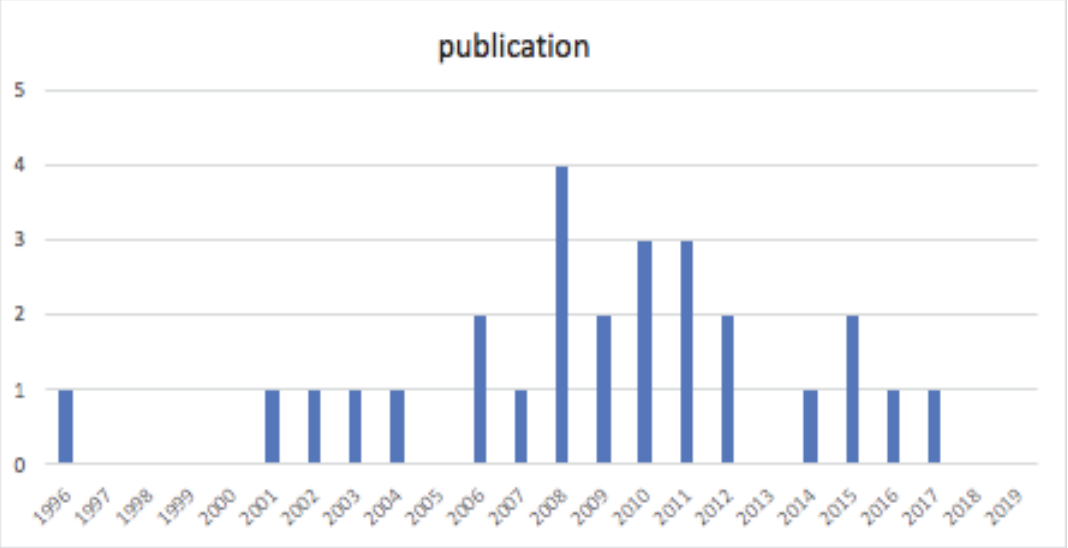

Figure 4.1: Publicate year distribution of articles

4.2.1 Categorization based on research method and study type

Based on the literature study of Runeson et al. [44] and Wierlinga et al. [45], 27 articles selected by us were classified according to the research method and the type of study adopted by each article, and the classification results were shown in Figure 4.2. The X-axis is the research method, and we have roughly divided them into six categories while the Y-axis is the type of study adopted in the article, which is divided into four parts: verification, evaluation, personal opinions and proposal of solution.

We found in these 27 articles,15 article adopts case study method which are P6, P8, P9, P11, P12, P13, P14, P15, P16, P17, P18, P21, P22, P23, P26, which exceeds the total number of half, but there are two (P10, P19) interview research method, there are three articles (P2, P20, P24) combine case study and interview as research method. In addition, 4 articles conducted empirical study (P1, P3, P4, P25), 2 articles adopted the framework (P7, P27), and only 1 article used the model (P5). Based on our research, most of the authors chose to study a specific project of a company or school, more than half of the literatures evaluated and verified previous existing studies, and only a few articles came up with some new methods and ideas.

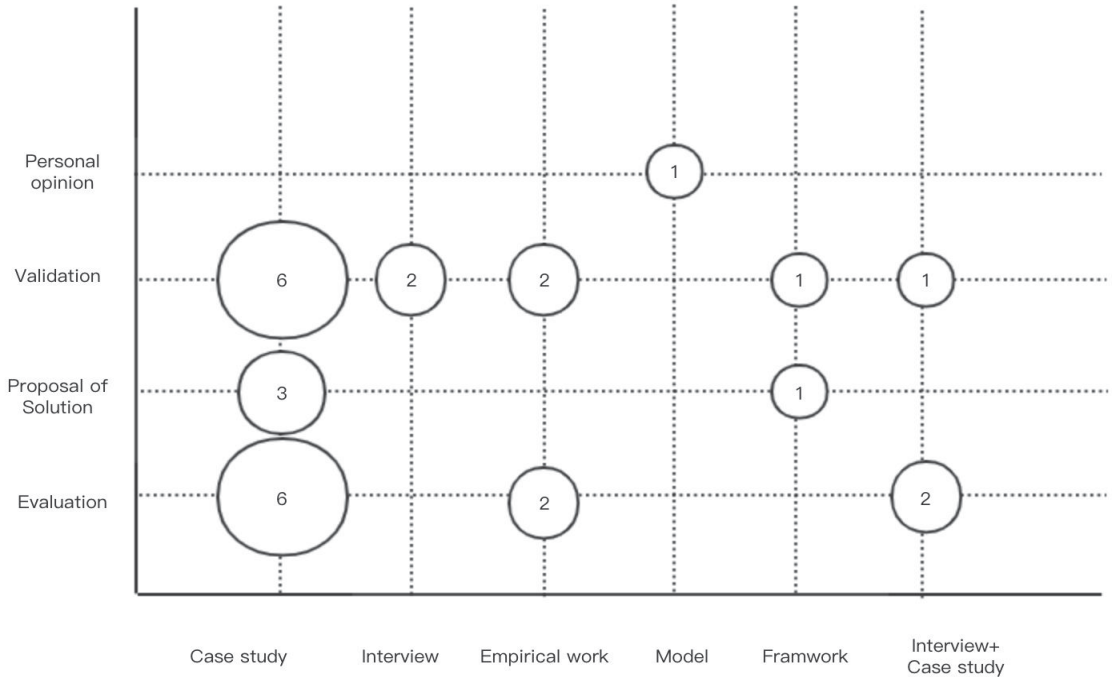

Figure 4.2: Categorization of studies

4.2.2 Quality assessment based on rigor and relevance

As Gorschek et al. [46] said, “Rigor refers to both how an evaluation is performed and how it is reported, one the other hand, Relevance refers to the potential impact the research has on both academia and industry.” Then we divided Rigor to three parts according to the guidelines [46] of evaluating literatures which are Context, Study Design, Validity/Limitation and divided Relevance to four parts which are Subjects, Context, Scale and Research Method for scoring the final selected 27 articles, the detailed scores of each part for each selected article are showing as in Appendix A. The quality assessment based on rigor and relevance are showing as visual representation chart as Figure 4.3. From Figure 4.3, we can clearly see that P2 has the highest rigor and relevance score while P10, P13, P19, P20 followed P12 and have relatively high scores, and P5, P22, P27 have relatively lowest rigor and relevance scores. Looking at the whole figure, 3 articles have high rigor scores, which are P2, P4 and P11, 6 articles have moderate rigor and relevance scores and 7 articles have relatively least rigor and relevance scores. On the other hand, as for relevance score, 20 articles have relatively high scores while 7 articles have least scores. Hence, we can get some much knowledge and thoughts for our research field from those articles that have high rigor and relevance scores.

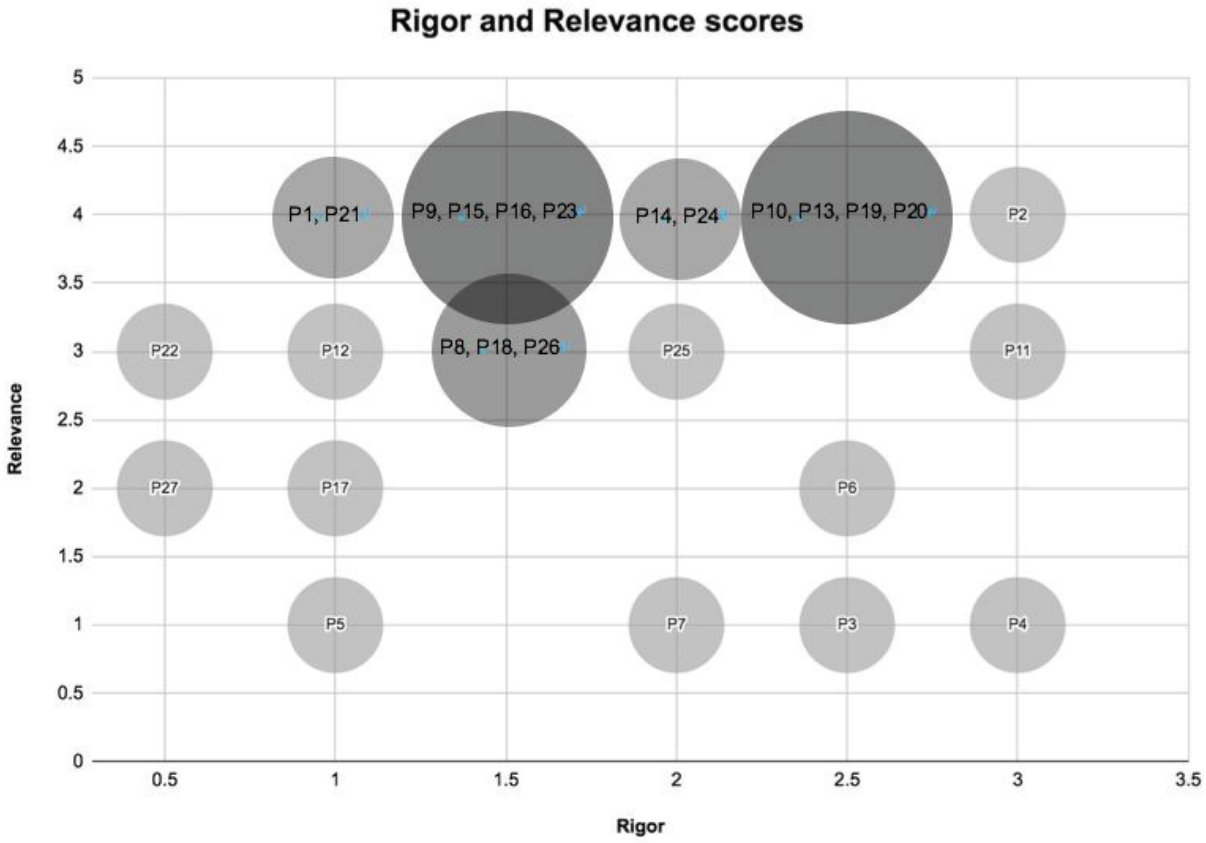

Figure 4.3: Rigor and relevance scores

4.2.3 Publication trends in requirement engineering literature regarding requirement communication

In order to better study the history and development trends of requirement communication, we conducted an analysis of publication trend for these articles. We used all the articles in the five iterations that we thought were related to requirement communication to conduct the publication trend research. Some of these articles were excluded after we or the supervisor read the abstract and the full text, but we still think that they are useful for studying the publication trend of article in the entire field of requirement communication. Their publication years range from 1996 to the present, we divided them equally into six groups, one for each four years, which is listed in Table 4.6.

From the data of Table 4.6, the number of articles that published each three year is shown intuitively and we can clearly see that more research on requirement communication started after 2004, perhaps because software engineering was less popular before 2000. In order to better see the development trend, we have made a line chart as shown in Figure 4.4. The figure shows that the number of articles on requirement communication has increased over the years. And the growth trend from 2015 is significantly faster than 2000. Although there is a small downward trend in the middle, we still believe that the overall trend is increasing.

Table 4.6 Publication trends

Year	Number of published articles
1996-1999	1
2000-2003	8
2004-2007	15
2008-2011	16
2012-2015	10
2016-2019	26

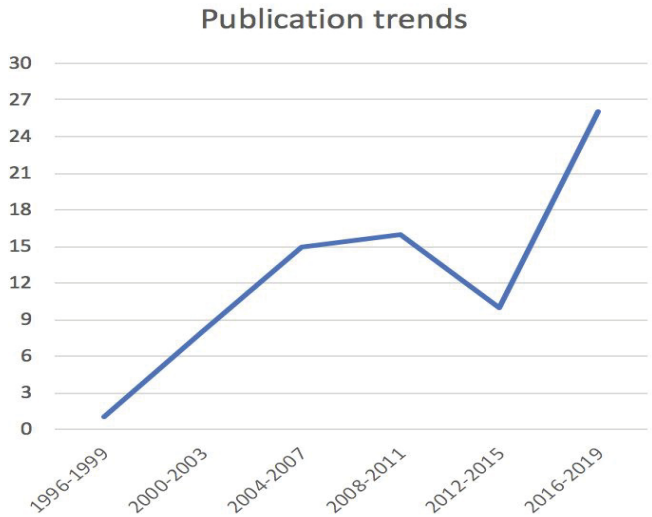

Figure 4.4 Publication trends

4.3 Analysis of SLR results

4.3.1 Contributions presented in the studies

The requirement engineering stage is a crucial phase as far as the holistic picture of the software development is concerned. There are many barriers involved which can be translated into structural issues thereby failing the efforts of all the stakeholders involved. P1 indicated three major communication barriers in the requirement engineering phase of software development. The research outlined the extent to which software practices are dependent on documentation as a communication medium suggesting the threats of the gap between two communities and suggests that informal communication should also be used to bridge that gap. People can pay attention to the specification documents so that they can be made more effective and software professionals must always demonstrate a willingness to cooperate.

In addition to this study, P2 regarded communication gaps as requirement communication challenges and then identified detailed causes of communication gaps from four aspects. The effects of communication gaps are also identified which are unclear requirement coverage to stakeholders, unclear requirement changes identification and so on. It is required to increase the understanding of gaps between causes and risks in requirements to easily point out potential gaps in the communication.

Following this line of approach, P16 also conducted a research in the pursuit of identifying the factors that influence communication media. Communication tools are used to access to communication media that is essentially helpful because of facilities of many to many and one to one communication modes. It has been evaluated that self-conception of bad language skills points to the inclination to practice text-based communication source. The study has explored multiple case study methodology researching eight developmental industries in software projects in two companies working in software industries. The results have suggested that text-based and audio associated communication media are preferably used.

The flow of requirement engineering heavily relies on the mode of communication which is used. The role of informal communication, for instance, meetings, calls is also studied by many researchers as they set out to improve the requirement engineering practices in various organizations. P5 provided a comparison of documented and informal approaches of communication and explored new opportunities for process improvement, requirement awareness and innovation of techniques and tools. The focus of the study approach was on quantitative flow models related to requirements flow in both solid and fluid representation while analyzing core aspects of visualization. Fluid information, using new FLOW model, displayed desirable outcomes regarding visualization aspects of informal communication to improve requirements engineering.

To effectively plan communication strategy, P6 regarding managing and planning communication channels in distributed development. The research proposed FLOW mapping systematic approach for managing and planning in distributed frameworks and concluded that FLOW management is desirable and sufficient to measure conformance and to plan communication strategy. The study proposes that future research must incorporate the analysis to improve the efficiency of conformance to reduce human efforts.

Few challenges are often observed in the global distribution of software projects. It has been observed that communication gaps such as bad management and inflexible software along with inappropriate communication modes can significantly affect the performance of software projects. A subsequent study P15 was again conducted to check the flow used for the improvement of communication requirements. The theoretical framework of this article is based on the global and universal theories of software engineering. The perspective specifically gives importance to communication gaps as it should be redressed through flow theory. Eventually, the study evaluated approaches to measure and evaluate the suggestions for future implications. The testing of these projects was completed in three zones, therefore, new strategies and implications should be introduced in globally distributed software projects.

Some researchers have also contributed to this strand of literature by analyzing the standardized artifacts and normatively accepted practices in the domains of requirement engineering. P8 introduced approaches to meet requirement engineering communication by a design science perspective and

prevented the requirement slipping by adding two transitional phases. The theoretical framework of this article is based on the probabilistic models that are employed by researchers to study the behavior of users and software designers and the impacts of marketing strategies. There are many factors that lead to the selection of the appropriate artifacts for requirements engineering, which are organizational culture and to match the frequency and intensity of the communication.

A second study in the same dimension was P10 in which the author studied many artifacts related to requirements communication including diagrams, user stories, and specifications to support several activities related to requirements. The research reflected that many artifacts are used to support different activities. The method used for research is to use clickable links that can add more content from concerning artifacts, and also, two artifacts are used together to operate with the same content.

Similarly, P18 was conducted to evaluate the proposal used in IT projects for the enhancement of user-developer communication. The research has suggested that user-developer communication approaches can significantly improve system quality and user acceptance. As it is understood that IT projects that are working on a large scale need essentially increased participation and high complexity, therefore, software development and technical specifications are required. The study has identified several trigger points to communicate, such as, variations in primary user needs as a means of communication.

In another dimension of requirement engineering, P17 considered a critical case study for user-developer communication process. Case study results have suggested researchers that potential benefits of consumer participation should be leveraged instead of taking it for granted. It is suggested for practical implications, to recognize user requirements to prioritize models accordingly in order to reduce communication gaps. It has also been observed that user participation is important in effective software systems. Disturbed and altered communication among developers and users is essentially important for the functioning of systems. Better communication can enhance large scale production and project achievement(s).

Agile development is specifically dependent on effective communication and feedback. P23 has utilized an impact on user communication on problems in software of agile development. Three case studies were analyzed and have evaluated that selection of communication methods is essentially important for the development of agile software. It has been observed that increased pliability on less enlightening communication channels outcomes in greater defect rates.

Many studies also recognized and acknowledged the significance of processes during the requirement engineering phases and analyzed the process dynamics so that specific goals and requirement corresponding to various themes can be met. P12 analyzed the comparison of requirements related to negotiation and hand-off. The paper measured requirements and design volatility and understanding of the architect's requirements during hand-off and negotiation. They also discussed the requirements types, which supports continuous product requirements engineering by differentiating four types of requirements.

Ineffective communication can significantly influence customers' wishes and it would be hard to run a project without proper assent. It has been found from P21, that handshaking can significantly improve proposal agreements, and negotiation can also outcome in better and effective communication. The

best working solutions should be reliable, cooperative and effective through handshaking and negotiation to get approvals and agreements for a project. Correspondingly, P9 studied characterization and identification of causes of requirements volatility while focusing on change analysis. The findings of the research revealed that variation in customer needs, increased understanding of the product by developers and changes in the policies of organizations are the main causes of requirements volatility. P13 also investigated the information which brokers related to requirements in social media and discussed a number of patterns of flow of information and many implications for requirements processes. The theoretical framework of this article is the concept of requirement-centric social network. The research questions related to research methods are related to different types of brokers, their consulting flows and dependent and independent requirements. The study investigated the presence of brokers and many ways in which these brokers influence information flow.

An extensive subdivision of the research related to requirements engineering is also associated with the recognition of new opportunities in this field. P4 carried out research on the effect of computer-mediated communication techniques on the requirement engineering and concluded that requirement negotiations are more effective by conducting discussions related to asynchronous structures prior to synchronous negotiation meeting. P11 studied the role of space and distances in requirements communication. The theoretical framework of these research findings relies on the theories of communication and interaction in an environment where successful products are the goals of peculiar software projects. They presented a case study of a project of software development and proved that RE distances impact project coordination and requirements communication. The results identified three categories of distances which include distances that can affect requirements communication, those which can indicate strong and weak alignment and those that characterize development models. Future work includes measurement of distances related to the artifacts and further finding ways to explore profiles and visualization of the distances.

P14 studied usage and choices of various communication tools in projects with the theory of Media Synchronicity. This theory strengthens the claims of media richness and it was found by many pieces of evidence that support the applicability of media synchronicity in choosing communication tools for GSD projects.

It has been observed that providing developmental feedbacks and visions to the team associates can significantly improve requirements engineering. The project plans and creating awareness among developers and users can improve performance as well. Awareness change is essentially important, as it was conducted in P27. The theoretical framework of this article is based on the visual representations of a requirement-centered social networks and associations among developers are required to be developed to improve communication and collaboration. Practical implications should be emphasized through awareness and requirement variations among contributors and developers.

Collaboration among team members is essentially required to be maintained in a project. The requirements should be developed in accordance with the user and project needs. P20 was conducted to identify the structure of need-driven interaction in shaping communication strategies. The study has analyzed that better communication and interaction between necessities and workers can significantly improve goal achievement processes of engineering projects. The study has evaluated that network analysis and better identification of needs and requirements are also helpful in shaping communication

strategies. Experts are needed to identify and evaluate the needs of users and projects to effectively develop communication approaches for a successful project.

Some studies also propagated various solutions which can be incorporated to the contemporary practices so that barriers in the smooth functioning of requirements engineering can be eliminated. P3 is based on a study that uses synchronous text-based communication to study requirement engineering for effective communication. The study concluded that CMC (computer-mediated communication) elicitation is better technology than CMC negotiation. The potential improvements arising from the relationship between designers and users of software development were also studied. P7 came up with the literature regarding effective communication in requirements elicitation and compared different methodologies. The proposed four-dimensional framework containing four different methodologies to promote a closer relationship among designers and users.

The most communication, highly critical between software engineering requirements, is software development and communication among workers. An empirical analysis was performed in P25 to assess means of communication for negotiation. Communication mode, face-to-face, was the most effective way to satisfy customers and to negotiate among consumers. The practical implication for better performance needs to be updated as good communication mode, such as, one-to-one communication. The product management collaboration and assess concerns of market and product development are the essentials of software development. The technological aspects should be considered for the growth of requirements engineering. New models and communication means should be addressed, evaluated and assessed to see how collaborative an organization is to promote its products. Media richness and media selection theories are used in P26. It is important for practical implications to orient and establishes goal-oriented and effective communication modes. The project of software and IT significantly depend on effective and efficient engineering knowledge and communication.

Finally, it can be reinstated that it is significantly important for an engineering project to be managed, especially software development systems. If system requirements specification will fulfil its roles and collaboration would be effective, then interest and input from stakeholders would be greater. Collaborative and effective management is required for a commercial project as well as a tool that enables software requirements. The chief requirement of a highly dynamic software is its support and flexibility. P22 described that collaborative software is essentially required for the requirements engineering. Theories deployed in this research chiefly account to system. Additionally, the dispersed plans are so intricate that nobody from team associates can perhaps own all the information about the plan, independently. P24 was conducted to identify the means of communication among developers and testers. The theoretical framework of this article is fundamentally related with agile testing of the software development. Various companies have observed several global software problems particularly in agile testing. Depending upon the task and target, means of communication need to be changed as present communication among testers through written communication ways are not found as effective.

4.3.2 Analysis of literatures regarding different roles involved in requirement communication

Most of the identified papers discussed the role of developer in requirement communication and they are more focused on the communication process between developers and users, which are studied by P16, P18 and P19. P1 discussed a communication framework for designers and users, P24 identified the communication method between developers and testers while P25 includes the requirement communication among clients and engineers.

We will introduce the different roles involved in each article as the following:

P7- This article aims to introduce a four-dimension framework including four different methodologies to promote a closer relationship among designers and users.

P16- This article divided stakeholders involved in requirement communication into two parts, which were technical team members and non-technical team members, and further discussed the most suitable appropriate communication method for each type of team member.

P17- The study conducted by a critical case study for user-developer and identified how the project failed despite the availability of user involvement at a higher level during the communication process.

P18- This study was conducted to evaluate the proposal used in IT projects for the enhancement of user-developer communication. The research suggested that the selection of communication approaches among users and developers can significantly affect system quality and user acceptance.

P19- It has been observed disturbed and altered communication among developers and users will affect the functioning of systems. And better communication among developers and users can significantly enhance large scale production and project achievement.

P24- The article focused on identifying the means of communication among developers and testers for especially Agile testing in worldwide distributed software development.

P25- This article proposed that it is necessary to select appropriate communication techniques or tools to support the communication between different stakeholders, i.e. clients and engineers during the product development, and an empirical analysis was performed to assess means of communication for negotiation.

4.3.3 Volatility in requirement communication

In this section, we will analysis the literatures regarding requirement volatility. Despite there are only four papers which are P9, P12, P20 and P27 including requirement volatility, they stated requirement volatility from different aspects.

We will introduce different aspects of requirement volatility:

P9- This article conducted change analysis for identifying the causes of requirements volatility, which were “variation in customer needs, increased understanding of the product by developers and changes in the policies of organizations”.

P12- The research paper measured requirements and design volatility and the degree of understanding of architect's requirements by receivers during hand-off and negotiation.

P20- This article proposed a pattern of functional-interaction for different functional teams and they found that it is important for team members included in functional team to receive the notice of requirement changes timely.

P27- This article asserted that the requirement changes should be informed to each member that involved in designing, developing and other contribution work. And a diagram is proposed to convey requirement changes to each team member that are dealing with the same requirement.

Those four articles help us understand requirement volatility to a large extent, but only P27 talked about how requirement changes convey within team members and it illustrated that awareness of requirement change is essentially important among stakeholders in requirement communication process. And P20 identified the importance of receiving requirement changes timely by team members.

4.3.4 Challenges encountered in requirement communication

In this section, we identified the challenges of requirement communication, and also extracted the factors or causes that lead to communication gaps during requirement communication process, which are separately introduced as following:

P1- This article concluded and indicated three major communication barriers in the requirement engineering phase of software development, which are shown as below:

- Ineffectiveness of the communication channels
- Organizational and social barriers
- Restrictions imposed by notations on expressiveness

P2- This study identified four factors that affect the requirement communication namely, temporal aspects, scale, decision structures and common views, and described causes corresponding to these four factors, which are shown as below:

- Size and complexity of the software development
- Weak understanding of other roles and corresponded responsibilities
- Period of provide requirements and corresponding changes is not sufficient
- Weak communication due to weak common visions or goals for software development

P8- This article studied that the standard of selecting communication artifacts can effectively reduce requirement slipping and also identified a key factor that affect the selection of the appropriate artifacts for requirements engineering, which is:

- Organizational culture

P10- This article proposed a kind of challenge in requirement communication and stated that some of the user representatives are observed to experience struggle while talking the requirements in an abstract way:

- Requirement mapping in an abstract way

P11- This article presented a case study of a project of software development and proved that RE distances impact project coordination and requirement communication. The results identified three

categories of distances, one of which include distances that can affect requirements communication, and the detailed descriptions for requirement communication are as below:

- Geographical distance: Which result in delaying the acquisition of requirement
- Organizational distance: Long distance and complex architecture between organizational units leads to weak communication and delay in obtaining information when transferring the information
- Psychological distance: Smoothly communication within a team and have a common view of sharing information
- Cognitive distance: the following challenges indicate that they belong to the previous challenge category

  Domain knowledge: The distance of domain knowledge may lead to the failure of catching useful information within team members

  Technical skill: For example, requirement engineers can help testers do testing if they have the technical skill of testing

  Process and Organizational knowledge: Increase this kind of knowledge through attending more team activities, and it is good for sharing regular information within team members.

  Priorities for system: The differences of priorities of a product identified by different roles.

- Adherence distance: the following challenges indicate that they belong to the previous challenge category

  Between delivered versus agreed requirements

  Agreed versus documented requirements
- Semantic distance

P15- This study has been observed that communication gaps that can significantly affect the performance of software projects, and the factors that lead to communication gaps are shown as below:

- Bad software management
- Inflexible communication software
- Inappropriate communication modes

P16- The study has explored multiple case study methodology researching eight developmental industries in software projects in two companies working in software industries and concluded several challenges to communication.

- Geographical
- Cultural
- Language distances

P17- The case study has identified that communication lapses that can significantly influence user-developer communication procedures, and the cause of the communication laps is as following:

- Researchers not regard the potential benefits of consumers as their own business

P23- This article identified a challenge encountered in requirement communication during development iteration:

- The selection of communication method

P24- Depending upon the task and target, the article proposed a challenge that will affect the effectiveness of requirement communication:

- The means of communication are changed

P25- The article proposed that is a kind of factor that affect effective requirement communication and the satisfaction level of consumer:

- The selection of communication techniques or tools

We then formulated a table to organize the challenges that identified in the articles , which is shown as Table 4.7.

Table 4.7: Identified challenges from the selected studies

ID	Challenge	Related articles
C1	Geographical and Culture	P1, P2 , P8, P16
C2	Organization (organizational architecture, relationship within organization)	P1, P2 , P11
C3	Cognitive distance (domain knowledge, technical skill, process and organizational knowledge, priorities)	P2, P11
C4	Psychological distance	P11, P17
C5	Adoption of professional language	P1
C6	Selection of communication artifacts, tools or methods	P1, P15, P23, P24, P25
C7	Size and complexity of the software development	P2
C8	Requirement mapping in an abstract way	P10

4.3.5 Solutions to improving requirement communication

In this section, we focus on some improvements for requirement communication proposed or organized by literatures, then identify and summarize the solutions to improving requirement communication.
Here we introduce the specific methods mentioned in each article:

P1- The research outlined the extent to which software practices are dependent on documentation as a communication medium suggesting the threats of the gap between two communities and suggested a solution should also be used to bridge that gap:

- Informal communication

P2- It is required to point out the potential communication gaps by increasing the understanding the causes and risks in requirement communication. The authors also suggested different aspects to reduce the challenging factors for effective communication.

- Increasing the understanding the causes and risks in requirement communication
- Software development models (waterfall and agile)
- Different organizational setups

P4- The solution which lead to requirement negotiations are more effective was carried out in this research:

- Conducting discussions within a development team related to asynchronous structures prior to synchronous negotiation meeting

P5- This article illustrated that fluid information, using new FLOW model, displayed desirable outcomes regarding visualization aspects of informal communication to improve requirements communication:

- New FLOW model

P6- This article is regarding managing and planning communication channels in distributed development and proposed an approach to manage communication effectively:

- FLOW mapping systematic approach

P7- The authors proposed a four-dimensional framework containing four different methodologies to promote a closer relationship and better communication among designers and users, which the four methodologies are shown as below:

- Four-dimensional framework :
  MUST,
  Joint Application Design,
  ULRC (User-Led Requirements Construction)
  SSM (Soft System Methodology)

P8- The study introduced and evaluated a model to meet RE communication by a design science perspective and prevented the requirement slipping by adding two transitional phases.

- A conceptual model of categorizing communication artifacts, metaphors, and levels of interaction that would be likely to correspond to the different phases of the RE process.

P10- The article proposed that work should be done in order to improve requirement communication by the following solution:

- Improve the facilitation of requirements mapping and transfer it from abstract way into an understandable form

P15- The study was conducted to check the flow used for the improvement of communication requirements. The perspective specifically gives importance to communication gaps as it should be redressed through the flow theory.

- FLOW theory

P16- The article proposed two different communication modes for different team members:

- Text-based and audio associated communication media is used for the members with bad language skill and in technical department

P17- There are communication lapses that can significantly influence user-developer communication procedures. It is suggested some solutions for practical implications, to accordingly in order to reduce communication gaps. Understanding participants and their needs to discuss designs of the models and how it will influence users, should be considered.

- User needs recognition and requirements prioritization model

P20- The study has evaluated that network analysis and better identification of needs and requirements are also helpful in shaping communication strategies. The authors also proposed a pattern of functional-interaction within functional team.

- Communication patterns of Interdependent Requirements-driven

P21- It has been found from a study conducted in this article that handshaking can significantly improve proposal agreements, and negotiation can also outcome in better and effective communication. On the other hand, handshaking is a kind mode of two-way communication process,

which means handshaking with the three phases of implementation proposal is a effective suggestion for project managers and development team to conduct requirement communication.

- Handshaking with implementation proposal

P23- In this article, three case studies were analyzed and have evaluated that selection of communication methods is essentially important for the development of Agile software, and one communication method is identified for improving requirement communication.

- Face-to-face communication method
- The selection of effective and flexible communication software

P25- Students were used as subjects for software requirements engineering and the study evaluated three different communication modes (Text-Based Communication, Face-to-Face, and Rich Media) are effective in satisfaction of consumers, then one of them was identified as the most effective way to communicate:

- Face-to-face communication method

P26- New models and communication means should be addressed, evaluated and assessed to see how collaborative an organization is to promote requirement communication among product management and a development team. The article was conducted for practical implications to establish a goal-oriented systems model for effective communication.

- Goal-oriented systems mode

P27- "A visual representation called a requirements-centered-social-network diagram" is proposed to convey requirement changes to each team member that are dealing with the same requirement, it has been also observed that social networks and associations are required for improving requirement communication among developers.

- Requirements-centered-social-network diagram

We then formulated a table to organize the solutions that proposed to improve requirement communication in the articles , which is shown as Table 4.8.

Table 4.8: Identified solutions from the selected studies

Solution type	Solutions	Related articles
Mode & Pattern	Interdependent Requirements-driven communication pattern	P20
	Handshaking with implementation proposal	P21
	Goal-oriented systems mode	P26
Model	Software development models (waterfall and agile)	P2
	New FLOW model	P5
	FLOW mapping systematic approach	P6
	A new conceptual model	P8
	User needs recognition and requirements prioritization model	P17

	FLOW theory	P15
Framework	Four-dimensional framework with four different methodologies	P7
Diagram	Requirements-centered-social-network diagram	P27
Communication method	Informal communication	P1
	Text-based and audio associated communication	P16
	Face-to-face communication method	P23, P25
	Effective and flexible communication software	P23
Suggestion	Increasing the understanding the causes and risks in requirement communication	P2
	Different organizational setups	
	Conducting discussions within a development team related to asynchronous structures prior to synchronous negotiation meeting	P4
	Improve the facilitation of requirements mapping and transfer it from abstract way into an understandable form	P10

5. Results and analysis of survey

In this research, we conducted and published our online questionnaires through a software called "wjx"[1], and sent the questionnaire link on the common social networking platforms, including Facebook, CSDN, and the software that are widely used in China, i.e. WeChat, QQ.

We finally received 132 responses of our online survey, and 15 responses are invalid responses because they didn't answer all of the questions, thus the data of 117 responses can be used to analyzed, which include 82 responses from software engineering practitioners and 35 responses from end-users. Therefore, the completion rate of our survey is 89%, and the number of responses from software practitioners is account for 70% of the number of valid questionnaires while the number of responses from end-users is account for 30% of the number of valid questionnaires, the following Figure 5.1 can indicate the rate of the number different kinds of respondents. Based on the article of Kitchenham et al. [47], the completion rate of valid questionnaires is sufficient for us to further analysis the data which from two different kinds of respondents. Moreover, we will conduct a detailed analysis based on the data sources of the basic identity information of two different kinds of respondents in the next part.

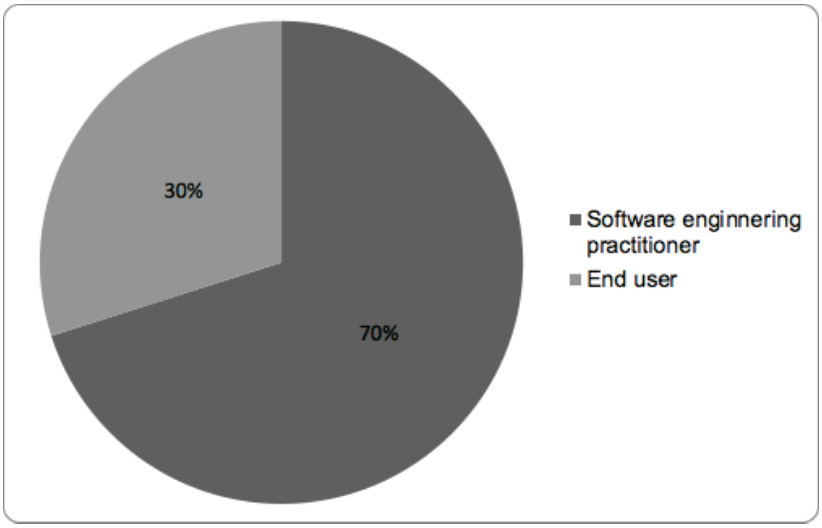

Figure 5.1: Percentage of respondents

5.1 Descriptive statistics of survey respondents

As the respondents we published the questionnaires to are from different company fields and have different work positions, we collected the relevant data and analysis their basic information, then the data of their general information can be generated clearly.

As for Question 1 in the questionnaire, we firstly divided the respondents into two classifications, i.e. software practitioners and end-users. Figure 5.2 shows the exact number of each kind of respondents. From Figure 5.2 we can clearly see that most of respondents are software practitioner.

[1] WJX is a platform, which focus on providing users to design online questionnaire survey, publish questionnaires to respondents, collect data of survey and the services of analysis survey result. https://www.wjx.cn/

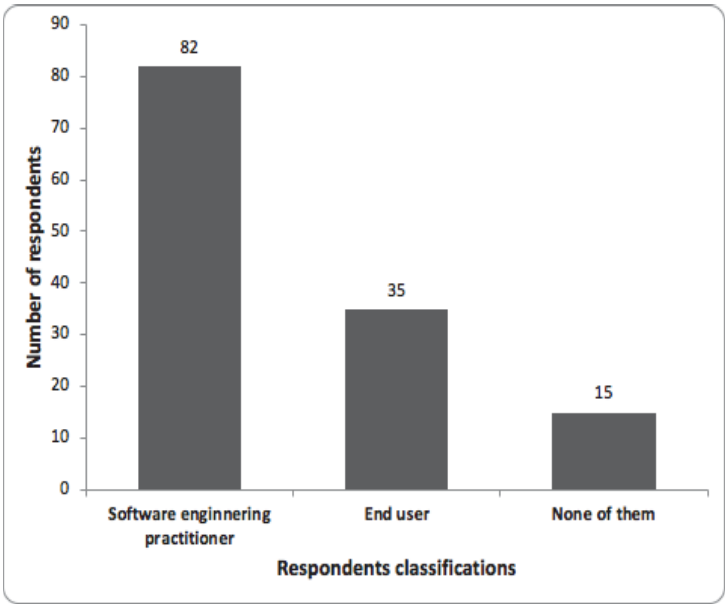

Figure 5.2: Classifications of respondents

Then we proposed Question 2 and Question 23 to figure out which company fields are the respondents belong to, which is shown as Figure 5.3. From Figure 5.3, we can clearly see that most of software engineering practitioner (51%) and end-users (46%) are belong to IT/Software development field, and the numbers of software engineering practitioner (16%) and end-users (6%) that belong to E-commerce field are relatively large. And the rest of respondents are from other company fields according to the following figure.

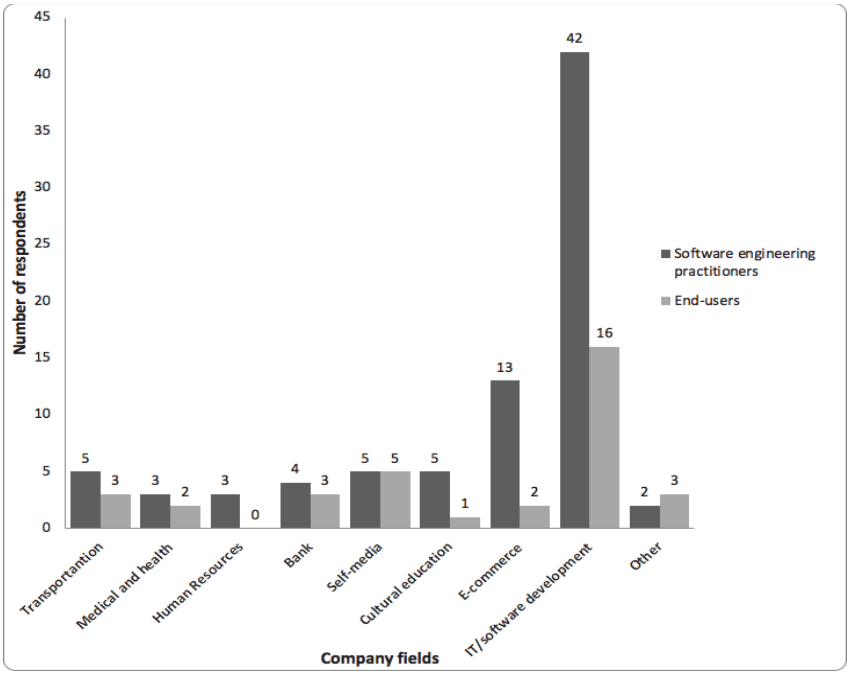

Figure 5.3: Company fields that respondents belong to

In Question 3 and Question 24, we focused on the knowledge level that respondents have, which are formulated as “Primary”, “Intermediate” and “Expert”. Regarding the definition of these three

options, we have not given specific requirements. All choices are based on the respondents' perception of their professional abilities, and the choice of this question is not our key research factor. The results are shown as Figure 5.4 that most of software engineering practitioners have intermediate (43%) and expert (39%) knowledge level of Software Engineering while just few of software engineering practitioners (18%) are novice of Software Engineering field, and the results of end-users are similar with software engineering practitioners, i.e. 49% intermediate and 37% expert level, which indicates that most of our respondents have relative high level of knowledge.

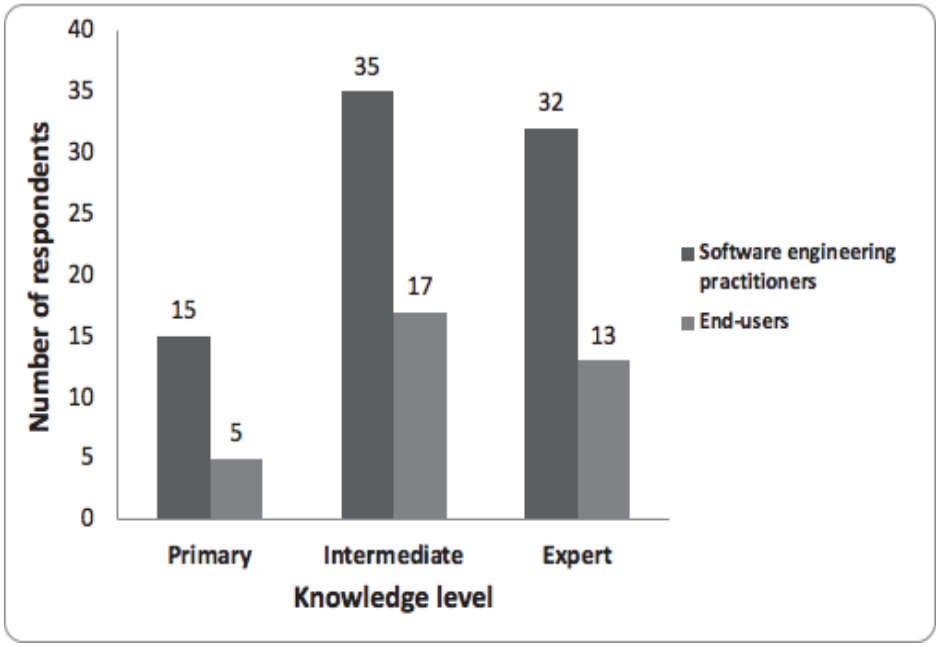

Figure 5.4: Knowledge level of respondents

For Question 4 and Question 25, we further focused on their work experience in Software Engineering field because we think their work experience may affect the communication methods they prefer to use in requirement communication. The results are shown in Figure 5.5. In the figure we can clearly see that most of respondents' work experience is in two ranges, which are 1-3 years and 4-6 years. 41% of software engineering practitioners have 1-3 year of work experience and 30% of them have 4-6 years of work experience. The number of software engineering practitioners that have less than 1 year (17%) and more than 6 years (11%) of work experience is relatively small. As for end-users, although most of them (49%) have 1-3 years of work experience, the percentage of less than 1 year (31%) of work experience is also high while some of them have more than 4 years of work experience. It can be concluded that among our respondents, most of software engineering practitioners have about 3-6 years of work experience, while end-users have relatively little work experience, which basically around 1-3 years.

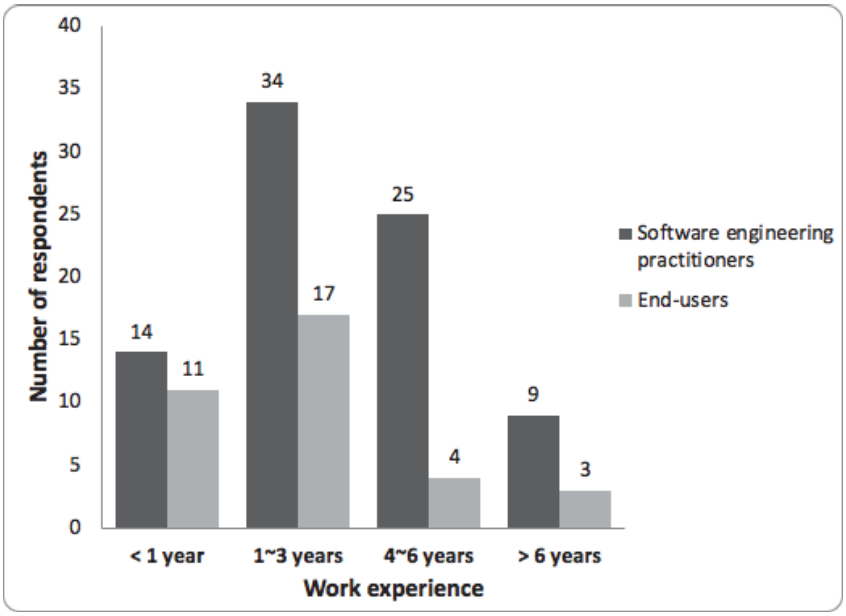

Figure 5.5: Work experience of respondents

In Question 5, it is formulated for software engineering practitioners. We think the roles identified in one development team may depend on the size of team. The results of Question 5 are shown as Figure 5.6. From Figure 5.6, we can see that 40% of software engineering practitioners are belong to the team which of size is 0~9 and 45% of software engineering practitioners are belong to the team which of size is 10~20. They can be concluded that most of respondents are belong to a relative medium scale of team size. And the types of roles in this medium size team are relative representative rather than those small teams or huge team size.

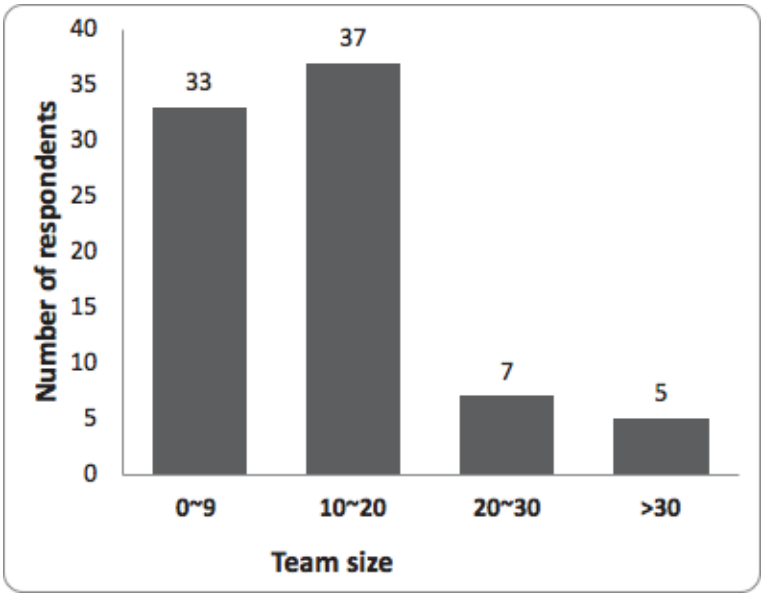

Figure 5.6: Respondents' team size

5.2 Roles in requirement communication

Although we know that participants will play many roles simultaneously in a software project, question 6 in the questionnaire only focuses on the general roles that play in their current software project team. The several roles we set as the options in this question are based on what we have learned from the overall introduction and knowledge of stakeholders in [57], from which we have selected some of the more common roles as options. And this question also targets to software engineering practitioners rather than end-users. The results can be shown as Figure 5.7. From the data we collected from the respondents, most of them are developers (51%) in the current team while 31% of software engineering practitioners are testers. There are few number of respondents belong to Project Managers (7%) and System Architects (5%) while some of respondents are belong to Requirement Engineer (13%). Then we can know that most of respondents' work positions are belong to development team.

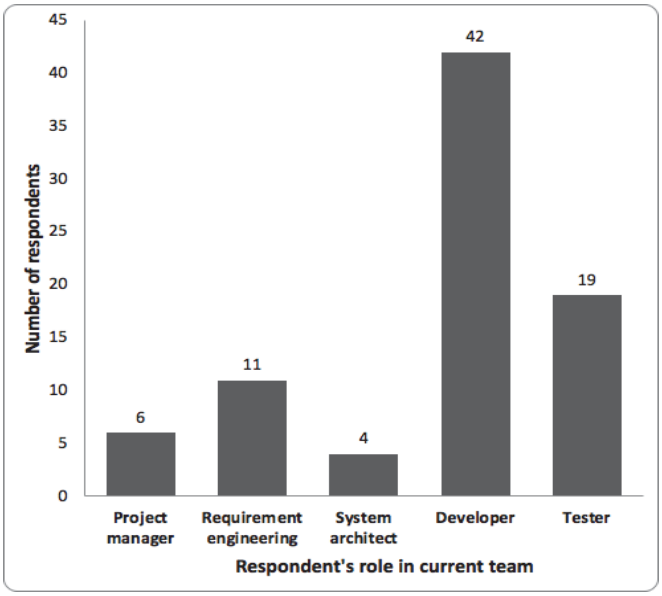

Figure 5.7: Respondents' roles in current team

According to Question 7, 8, 9, 10 and 11, the respondents will be forwarded to this question after their roles identifying through answering the Question 6. We identified some typical responsibilities corresponding to each role based on our own understanding and recognition, and according to the information of plenty of relevant literatures mentioned in systematic literature review. Maybe the responsibilities we identified didn't cover more specific responsibilities taken by each role, but responsibility identification is not the focus of our research, we just wanna identify the responsibilities that recognized and taken by most people. The mainly objective of formulating those questions is identifying the most common responsibilities of each role and filtering the responsibilities that relate to requirement communication process, which will further help us analysis the effectiveness of different of communication methods according to different responsibilities of roles. Then the results are shown as Figure 5.8, Figure 5.9, Figure 5.10, Figure 5.11 and Figure 5.12.

As for the role of Project Manager of those respondents, the results can be clearly shown in Figure 5.8. We can clearly see that the mainly responsibility of this role is "Management of the entire team" because all of 6 project managers selected this option, and the responsibility of "Supervision of time and budget of developing product" is relative important because 83% of project managers selected it. The rest of the other three responsibilities are relative important according to the data, i.e. 67% of project managers chose "Market research conduction and requirements mining" while 50% of project managers chose both "Product planning and designing in the early stage" and "Product spreading and promotion". In these options, the responsibilities that may involve requirement communication are "Management of the entire team" and "Supervision of time and budget of developing product". The percentage that chose these two options are also high.

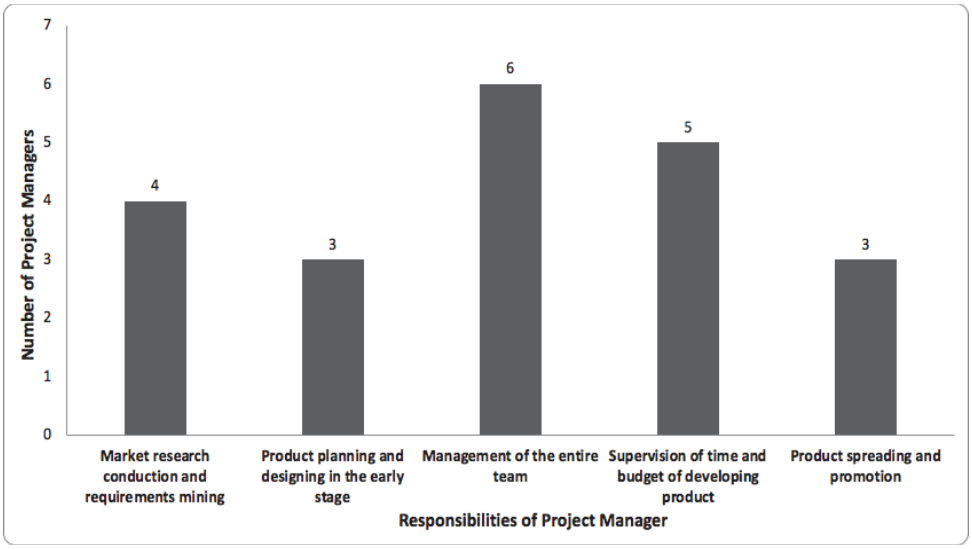

Figure 5.8: The responsibilities of Project Manager

As for the role of Requirement Engineer of those respondents, the results can be clearly shown in Figure 5.9. We can clearly see that the mainly responsibility of this role is "Requirements analysis, filtration and classification" because all of 11 requirement engineers selected this option, and 91% of requirement engineers selected "Writing requirement documentation" indicates that writing requirement documentation is the most important responsibility of requirement engineer. The responsibilities of "Identification and optimization of overall project requirements" (82%) and "Requirements documents updating according to requirement changes" (73%) are relative important while they are also related to requirement communication process. Although "Collection of user feedback and communicating with team members" (55%) may be also related to requirement communication, it is not chosen by most of requirement engineers.

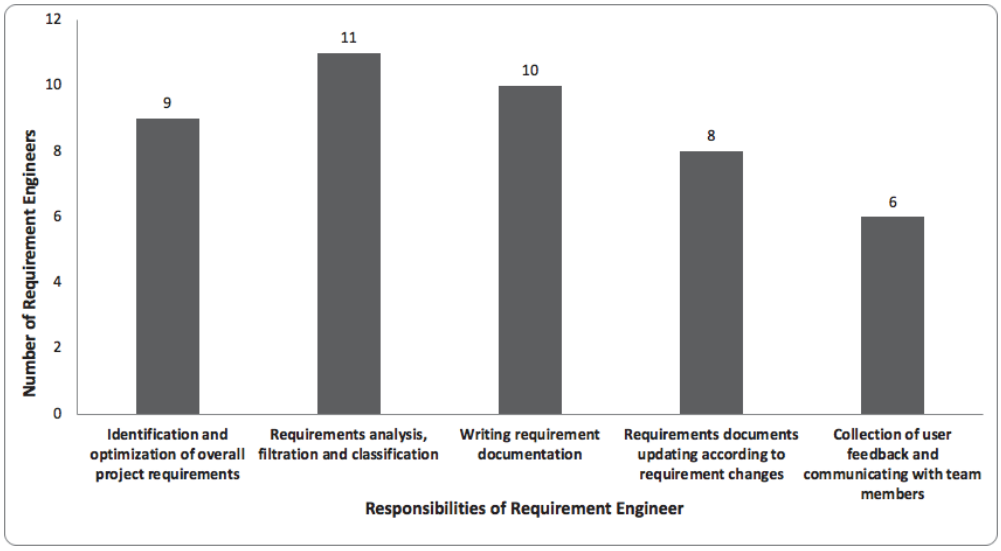

Figure 5.9: The responsibilities of Requirement Engineer

As for the role of System Architect of those respondents, the results can be clearly shown in Figure 5.10. We can clearly see that the mainly responsibilities of this role are "Overall software architecture formulation and system decomposition vertically and horizontally" and "Coordinating and communicating with all developers" because all of 4 requirement engineers selected these two options, on the other hand, those two responsibilities may relate to requirement communication process, which means system architects sometimes play a really important role in development team. Although 50% of system architects selected "Accurate requirements identification according to communicating with requirement engineer", it is also related to the process of requirement communication. What's more, 75% of system architects selected "Technical selection and negotiation with project manager" means it is a relative important responsibility. And few (25%) of them chose "System feasibility verification and technical risks assessment" may indicate that it is a common responsibility of system architect.

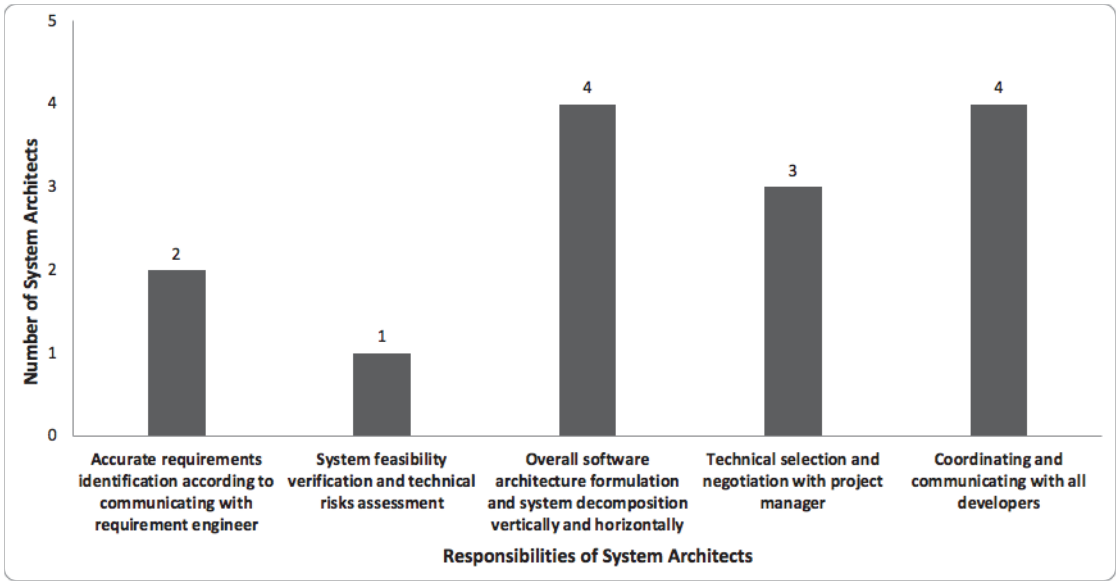

Figure 5.10: The responsibilities of System Architects

As for the role of Developer of those respondents, the results can be clearly shown in Figure 5.11. We can see that most of our respondents are developers according to Figure 5.7. The mainly responsibilities of this role shown in Figure 5.11 are "Software programming and coding" and "Keep communicating with team members" because 95% of developers selected these two options while the latter is related to requirement communication process. Although only 24% of developers selected "Assistance of testers to complete software system and module testing" which means few of them have this responsibility in the current team, it is also related to the process of requirement communication. "Conduction of software unit tests" (36%) is also not a really common responsibility of developer, and "Writing related technical documents" (50%) is more common than the former. What's more, 90% and 83% of developers selected "Functional designing and interface beautification" and "Assurance of the stability and reliability of system" means they are relative important responsibilities in software engineering process.

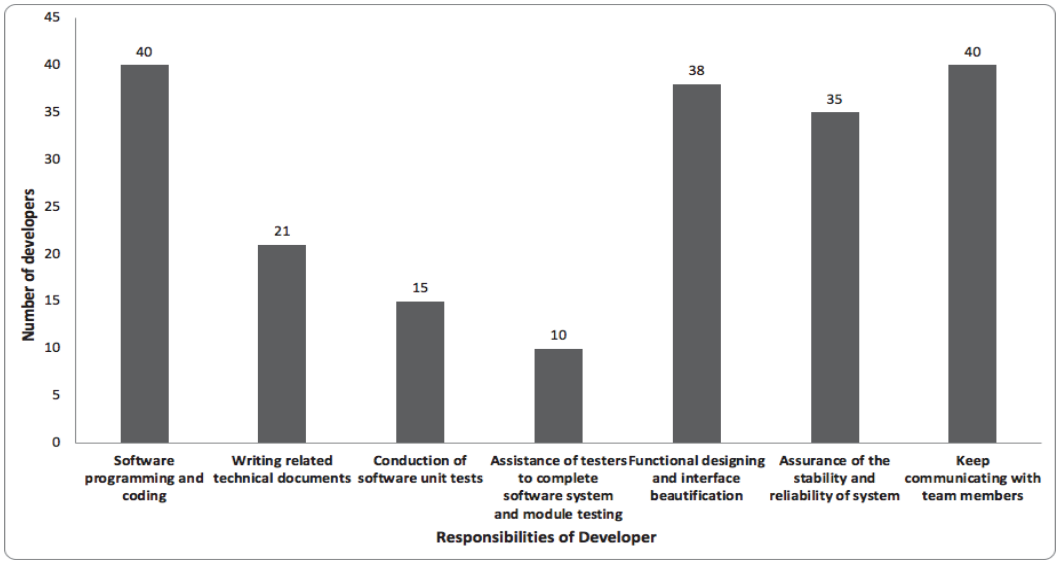

Figure 5.11: The responsibilities of Developer

As for the role of Tester of those respondents, the results can be clearly shown in Figure 5.12. We can see that the most common responsibility is "Communicate about testing results with developers" because all of 19 testers selected this option, on the other hand, this responsibility is also related to requirement communication process. Most of testers selected "Conduction of system functional testing" (95%) shows that it is a relative common responsibility of testers. A relatively large number of testers always do "Recording test results and writing test reports" (79%). Some of testers do "System defects tracking and analysis" (63%) and "Peer review of test cases" (68%) while the latter is related to requirement communication process. Approximately half of testers selected "Writing test cases according to the content of the project requirements" (53%) and few of them do "Test environment establishment" (42%).

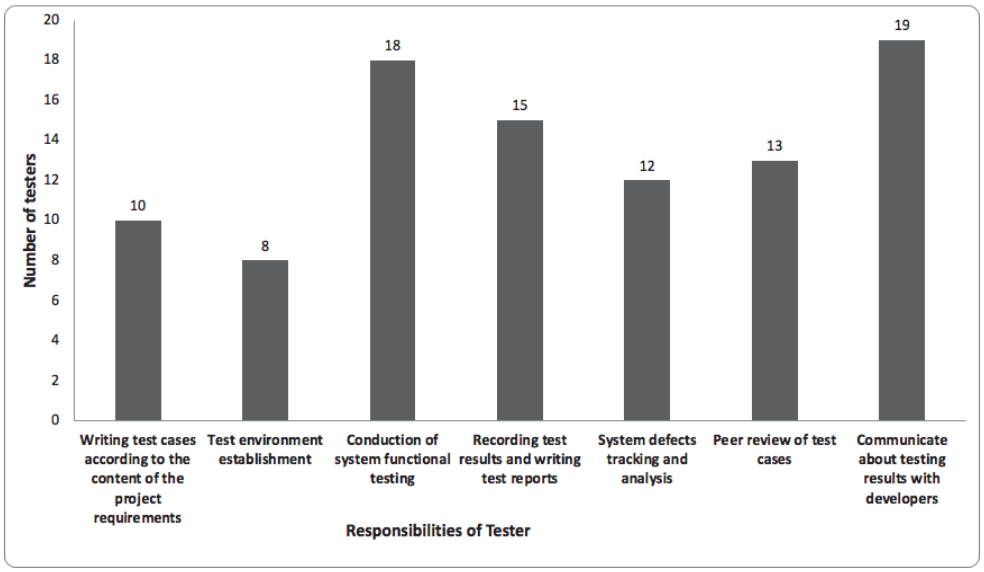

Figure 5.12: The responsibilities of Tester

As a kind of role, we formulated this open question, i.e. Question 37 for end-users in order to know their responsibilities that involved in requirement communication process. From the results we collected from end-users, we first excluded 5 answers that are invalid, then we found that 15 of them proposed a kind responsibility of "provide feedback about product final delivery", which means half of end-users think it is a mainly important responsibility as an end-user. 10 of them proposed the answers that relate to one kind of responsibility of "provide accurate requirements to the development team" while 6 answers talk about the responsibility of "update requirement changes to the development team in time". Therefore, the mainly responsibilities related to communication process of end-users according to the answers of questionnaires can be concluded as: provide feedback about product final delivery (43%), provide accurate requirements to the development team (29%) and update requirement changes to the development team in time (17%).

Question 12 is a question for software engineering practitioners. We asked what communication methods they prefer to use. The two most selected methods are face-to-face meeting and writing requirement documents, with 39% and 27% of practitioners choosing them respectively. In addition, 18% of practitioners chose E-mail, 9% chose video meeting, and only 7% chose voice meeting.

Question 26 is for end users, in which we ask the same question as question 12. The most popular option is also face-to-face meeting, chosen by 43% of end users. In second place was video meeting, with 37% of end users saying they preferred it. In addition, 11% of participants opted for E-mail, 6% for voice meeting, and only 3% for writing requirement documents. We asked two types of participants the same question in questions 12 and 26, but with different probabilities of being chosen the same method, so we decided to discuss whether their choices were related to individual roles.

We also combined the answers to questions 12 and 26 with the answers to questions 4 and 25 to summarize the data shown in the Table 5.1 below. We can see clearly the different work experience of the participants on the choice of the communicate method is also different. For example, software engineering practitioners with less than one year of experience prefer face-to-face meetings, and for more than 6 years working experience in software engineering practitioner, they prefer to choose E-mail and write the requirement document. A similar distribution of data exists for end users.

Therefore, we decided to discuss whether the working experience of the participants affected their choice of the requirement communicate method.

Table 5.1 The distribution of the communicate method of the participants' work experience

Roles	Work Experience	Face-to-face Meeting	E-mail	Voice Meeting	Video Meeting	Writing requirement document
Software engineering practitioner	Less than 1 year	5	2	1	2	4
	1-3 years	16	8	1	3	6
	4-6 years	10	2	2	2	9
	More than 6 years	1	3	2	0	3
End user	Less than 1 year	2	0	0	9	0
	1-3 years	10	2	2	3	0
	4-6 years	2	2	0	0	0
	More than 6 years	1	0	0	1	1

The relationship between the choice of communicate method and the participants' role.

We classified the participants according to the answers to questions 1 and 6, and divided them into: project manager, requirement engineering, system architect, developer, tester and end user. We conducted the chi-square test to analyze whether there is a relationship between the role of the respondents and the communication method they chose.

We assumed the H_0 as: "There is no significant relationship between respondents' roles and their choice of communicate method". On the contrary, the H_1 was: "There is a significant relationship between the two variables".

As shown in the Figure 5.13, we can find that chi-square value =26.018[a], df=20, p=0.178[b] >0.05. So the H_0 should be accepted. However, 21(70%) cells have expected count less than 5, which is a violation of chi square test conditions. In this case, we couldn't use the chi-square test, so we did the Fisher's exact test. The Figure 5.13 shows that the Fisher's exact test value is 27.111 and p value is 0.06 >0.05. So we can accepted H_0 that there is no significant relationship between the choice of communicate method and the participants' role.

Participant_Role * Communication_method Crosstabulation

Count

		Communication_method					
		E mail	Face to face	Video meeting	Voice meeting	Writing requiremeng documents	Total
Participant_Role	Developer	8	16	2	3	13	42
	End user	4	15	13	2	1	35
	Project manager	1	2	1	1	1	6
	Requirement engineering	3	4	1	0	3	11
	System architect	0	2	1	0	1	4
	Tester	3	8	2	2	4	19
Total		19	47	20	8	23	117

Chi-Square Tests

	Value	df	Asymptotic Significance (2-sided)	Monte Carlo Sig. (2-sided)		
				Significance	99% Confidence Interval Lower Bound	Upper Bound
Pearson Chi-Square	26.018[a]	20	.165	.178[b]	.134	.222
Likelihood Ratio	29.298	20	.082	.156[b]	.114	.198
Fisher's Exact Test	27.111			.060[b]	.033	.087
N of Valid Cases	117					

a. 21 cells (70.0%) have expected count less than 5. The minimum expected count is .27.

b. Based on 500 sampled tables with starting seed 92208573.

Figure 5.13: The chi-square test result of participant roles vs the communicate method

The relationship between the choice of communicate method and the participants' work experience

In order to analyze whether the working experience of participants has an impact on their choice of communication method, we carried out chi-square test from the working experience of software engineering practitioners and the experience of end users.

First, we assume that H_0 is: "There is no relationship between the choice of communicate method and the software engineering practitioners' work experience." In contrast, H_1 is: "There is a significant relationship between these two variables." The following Figure 5.14 is the result of a chi-square test of the software engineering practitioner's work experience and the data of the selected communicate method. We find that H_0 is acceptable because Chi-square test value = 12.569[a], df = 12, p = 0.408[b]> 0.05. However, there are 14 (70%) cells with an expected count lower than 5. In this case, we perform Fisher's exact test as a supplement. Fisher's exact test value is 12.591, and p is 0.316[b]. To summarize, there is no significant relationship between the choice of communicate method and the participants' work experience.

ParticipantWorking_Experience * Communication_method Crosstabulation

Count

		Communication_method					
		E mail	Face to face	Video meeting	Voice meeting	Writing requiremeng documents	Total
ParticipantWorking_Experience	1-3 years	8	16	3	1	6	34
	4-6 years	2	10	2	2	9	25
	Less than 1 year	2	5	2	1	4	14
	More than 6 years	3	1	0	2	3	9
Total		15	32	7	6	22	82

Chi-Square Tests

	Value	df	Asymptotic Significance (2-sided)	Monte Carlo Sig. (2-sided)		
				Significance	99% Confidence Interval Lower Bound	Upper Bound
Pearson Chi-Square	12.569[a]	12	.401	.408[b]	.351	.465
Likelihood Ratio	13.416	12	.340	.474[b]	.416	.532
Fisher's Exact Test	12.591			.316[b]	.262	.370
N of Valid Cases	82					

a. 14 cells (70.0%) have expected count less than 5. The minimum expected count is .66.

b. Based on 500 sampled tables with starting seed 1436388411.

Figure 5.14: The chi-square test result of the work experience of software engineering practitioner vs the communicate method

We made the same test for the end user. We assumed that H_0 was: "There is no relationship between the choice of communicate method and the end users' work experience." Instead, assume H_1 was: "There is a significant relationship between these two variables." The result shows in Figure 5.15, the chi-square test value is 31.426a, df=12, p = 0.02 <0.05. However, there are 18 (90%) cells with a expected count lower than 5. We also did the Fisher's exact test as a supplement. Fisher's accurate test value is 22.576, p is 0.002, so we get the result that there is a relationship between the work experience of the end user and the choice of communication method.

EnderuserWorking_Experience * Communication_method Crosstabulation

Count

		Communication_method					
		E mail	Face to face	Video meeting	Voice meeting	Writing requiremeng documents	Total
EnderuserWorking_Experience	1-3 years	2	10	3	2	0	17
	4-6 years	2	2	0	0	0	4
	Less than 1 year	0	2	9	0	0	11
	More than 6 years	0	1	1	0	1	3
Total		4	15	13	2	1	35

Chi-Square Tests

	Value	df	Asymptotic Significance (2-sided)	Exact Sig. (2-sided)
Pearson Chi-Square	31.426[a]	12	.002	.004
Likelihood Ratio	26.373	12	.010	.003
Fisher's Exact Test	22.576			.002
N of Valid Cases	35			

a. 18 cells (90.0%) have expected count less than 5. The minimum expected count is .09.

Figure 5.15 The chi-square test result of the work experience of end users vs the communicate method

After investigating what communication ways our respondents prefer, we also formulated an open question, i.e. Question 13 and Question 27 in order to know other communication ways that we didn't mention previously. From the results of the questionnaire we collected, there are 8 software engineering practitioners and 7 end-users selected "yes" in this question. In the answers from software engineering practitioners, we found that there are 5 software engineering practitioners use Wechat to conduct requirement communication and 2 of them use Facebook to communicate requirements within their team. Among end-users, we found that there are 5 of them used Facebook to communicate requirements with the development team while one of them prefer to use QQ. In addition to that, there are 2 answers that not relate to this question, which are determined invalid answers.

According to the 4 common types of communication which are verbal, nonverbal, written and visual, we think the other ways that respondents proposed are belong to the type of written type of communication and these specific ways can be further classified to social media communication method. Therefore, all of valid answers from respondents to this open question are belong to social media communication method that we didn't mention in Question 12 and Question 26.

In questions 14 and 28, we asked the same question to software engineering practitioners and end users. We've listed five communicate methods and provided four levels of effectiveness. Considering that it is possible that the participants have not used the communication method, an option called "don't know" has been added.
And the participants could only choose one of the five levels for each type of method.

We summarize the answer data of software engineering practitioners in the Table 5.2 below, and it is clear that more than half of software engineering practitioners consider face-to-face meetings to be extremely effective. 56% of software engineering practitioners thought E-mail was effective, and similarly, 37% said writing requirement documents was effective. For voice meeting, 50% of participants chose it was slightly effective, and 29% chose it was useless. There were similar results for video, meeting, with 40% of participants choice video meeting was slightly effective and 21% was completely ineffective.

Table 5.2: The results distribution of the effectiveness of communication methods for software engineering practitioners

Communicate method	Not at all	Slightly effective	Effective	Extremely effective	No idea
Face to face meeting	2(2%)	10(12%)	20(24%)	50(61%)	0
E-mail	3(4%)	16(20%)	46(56%)	17(21%)	0
Voice Meeting	24(29%)	41(50%)	12(15%)	5(6%)	0
Video Meeting	17(21%)	33(40%)	21(26%)	11(13%)	0
Writing requirement document	2(2%)	22(27%)	30(37%)	28(34%)	0

The following Table 5.3 shows the answer data for the end user. It shows that 57% of end users think face-to-face meeting is extremely effective, 51% think E-mail is effective, and 54% think voice meeting is slightly effective, similar to the results of software engineering practitioners discussed above. But the difference is that nearly half (46%) of end users think video meeting is effective, which is much higher than the percentage of software engineering practitioners who choose it (26%). For writing requirement documents, extremely effective was the least frequently selected option, with more users saying it was only slightly effective. By comparison, we find that although they have different methods, the differences are subtle.

Table 5.3: The results distribution of the effectiveness of communicate methods for end users

Communicate method	Not at all	Slightly effective	Effective	Extremely effective	No idea
Face to face meeting	1(3%)	4(11%)	10(29%)	20(57%)	0
E-mail	1(3%)	6(17%)	18(51%)	10(29%)	0
Voice Meeting	6(17%)	19(54%)	7(20%)	3(9%)	0
Video Meeting	0	15(43%)	16(46%)	4(11%)	0
Writing requirement document	10(29%)	18(51%)	6(17%)	1(3%)	0

After the preliminary analysis of the data above, we believe that there are some differences in the effectiveness of communicate method between the two types of participants. Therefore, we calculated the average and variance of the effectiveness of each communication method. We assign a value to each degree of effective, "Not at all" is 1 point, "Slightly" is 2 points, "effective" is 3 points, and "Extremely" is 4. Since we didn't get the answer to "no idea", we didn't consider this one. We show the average and variance results for each communication method in the Table 5.4.

According to calculate the average and variance in the table, we can see that for all participants, the most effective method is face to face meeting, but for software engineering practitioners, the least effective way is to voice meeting, and for end users, the least effective method is to write requirement documents. For the two types of participants, the deviation of various communication methods is between 0.6 and 0.7, which also indicates that the score of respondents is different.

Table 5.4: The average and variance results of effectiveness degree of communication method

Communicate method	Average-Software engineering practitioner	Average-End user	Variance-Software engineering practitioner	Variance-End user
Face-to-face meeting	3.4	3.4	0.6	0.6
E-mail	2.9	3.1	0.5	0.6
Voice Meeting	2	2.2	0.7	0.7

Video Meeting	2.3	2.7	0.9	0.5
Writing requirement document	3	1.9	0.7	0.6

Based on the above statistics, we decided to analyze whether participants' assessment of the effectiveness of each communicate method was related to their roles. We still classify the roles of participants according to the previous method. The data were classified based on different communication methods, and then the chi-square test was carried out respectively.

We used the chi-square test to analyze whether there was any relationship between the role of the participants and the effectiveness of the assessed communication method. First, assume that H_0 was: "There is no significant relationship between the role of the participates and the effectiveness of the communication method". Instead, assume H_1 is: "There is a significant relationship between these two variables."

We present a chi-square test result in Figure 5.16, which is the result for face-to-face meeting. The chi-square value = 23.486[a], df = 15, p = 0.082> 0.05. However, 17 (70.8%) cells expected count less than 5, and the chi-square test requires that the expected count less than 5 should not exceed 20% of all cells. Therefore, we conducted the Fisher's extract test. The Fisher's extract test value is 24.511, and p value is 0.018 <0.05. So we can say H_0 can be accepted which means that the role of the participants is related to the effectiveness of the face-to-face meeting rating. We ran the same test on the data for each communication method and summarized the data in the following Table 5.5.

Participant_Role * FCF_Effectiveness Crosstabulation

Count

		FCF_Effectiveness				
		1	2	3	4	Total
Participant_Role	Developer	1	3	5	33	42
	End user	1	4	10	20	35
	Project manager	0	1	2	3	6
	Requirement engineering	1	1	6	3	11
	System architect	0	2	2	0	4
	Tester	0	3	5	11	19
Total		3	14	30	70	117

Chi-Square Tests

			Asymptotic Significance (2-sided)	Monte Carlo Sig. (2-sided)		
					99% Confidence Interval	
	Value	df		Significance	Lower Bound	Upper Bound
Pearson Chi-Square	23.486[a]	15	.074	.082[b]	.050	.114
Likelihood Ratio	23.799	15	.069	.072[b]	.042	.102
Fisher's Exact Test	24.511			.018[b]	.003	.033
N of Valid Cases	117					

a. 17 cells (70.8%) have expected count less than 5. The minimum expected count is .10.

b. Based on 500 sampled tables with starting seed 562334227.

Figure 5.16: The chi-square test result of the role of the participant vs the effectiveness of each communicate method

Table 5.5: The results of Pearson Chi-square and Fisher exact test

Communicate method	Chi-Square value	df	Fisher's Exact Test Value	Significance (p value)
Face to face meeting	23.486a	15	24.511	0.018b
E-mail	22.806	15	20.933	0.076
Voice Meeting	10.226	15	9.895	0.812
Video Meeting	31.226	15	29.868	0.001
Writing requirement document	51.224	15	45.576	<0.001

From the p values obtained from the chi-square test of these five methods, the effectiveness of face-to-face meeting, video meeting and writing requirement documents was related to the role of participants ($p<0.05$), while the effectiveness of E-mail and voice meeting was no significant relationship to the role.

5.3 Results on requirement types in requirement communication

As for Question 15, we decided to combine it with Question 29. Because the two questions are all researching the most common type of requirements that respondents have ever met. The results are shown in Figure 5.17. In this Figure, we can clearly see that most of software engineering practitioners received two types of requirements, which are “Functional requirements” (44%) and “Quality attribute requirements” (43%) while these two types of requirements are also the most frequently proposed by end-users, which of the corresponding percentages are 40% and 34%. It indicates that functional requirements and quality requirements are always involved in requirement communication process. Some of software engineering practitioners (10%) and end-users (14%) also selected “Use case requirements” illustrate that there are a few end-users who use the UML modeling standardized requirements expression to propose requirements. However, the type of data requirement (11%) is relative rarely proposed by end-users, we can roughly estimate that maybe end-users prefer to propose intuitive requirements (e.g. functional requirements) rather than data requirements, because for most end-users, they are more care about what the system or software can do, and what the final value of the product is expected, but they don’t know much about how to achieve this value, which means they are not familiar with the data involved in the requirements.

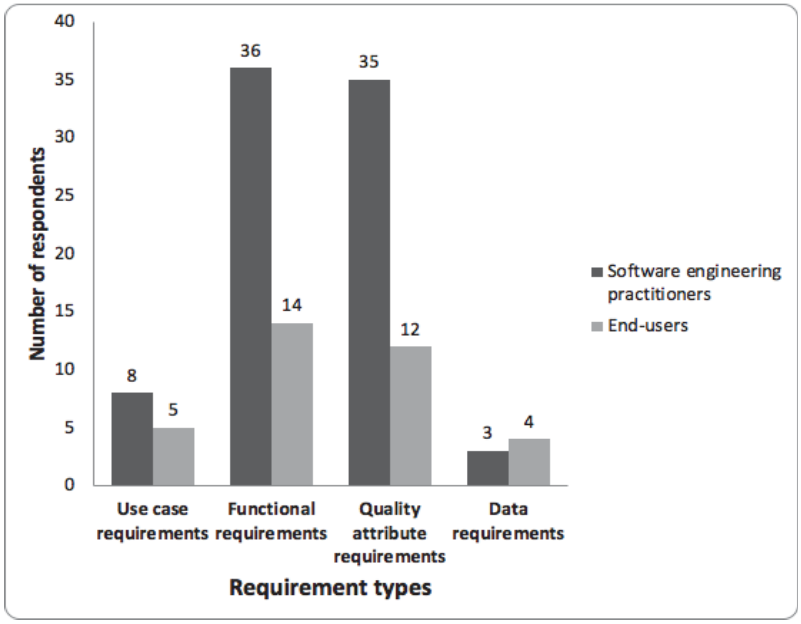

Figure 5.17: The most common type of requirements that respondents met

As for Question 16 and Question 30, we focused on whether the respondents adopt different communication methods based on different requirement types. From the answers of software engineering practitioners, 56 of them chose "Yes" and 26 of them chose "No". Among the answers of "Yes", 35 of them proposed that they prefer to use written type of communication method, e.g. writing requirement documents, Email because they prefer to describe use cases for system requirements with the clearly form of text. 20 of them will adopt email and some social media software to communicate functional requirements. And 20 of them will adopt visual type of communication method for quality requirement type because they think it is easier to communicate about the system attributes by face-to-face. 7 of them said they will adopt written communication type to communicate about the type of data requirements because the verbal type of communication method does not convey the basic data dimensional analysis of product requirements, thus, the requirement communication in written form is more suitable for data requirements.

From the answers of end-users, 30 of them chose "Yes" and 5 of them chose "No". Among the answers of "Yes", 10 of them adopt visual type of communication method for proposing quality attribute requirements while 10 of them said they will adopt the communication methods according to the selection of development team. 7 of them said that they will adopt written communication type, e.g. Email, Wechat for proposing functional requirement type. However, no end-user talks about the communication methods they will adopt for use case requirement type.

5.4 Survey results on handling requirement changes in requirement communication

In Question 17, we focused on the most common type of requirement changes that software engineering practitioners have received. On the other hand, we also survey the most common type of requirement changes that end-users proposed to software development teams in order to analysis Question 31. We further investigated whether the requirement changes have been achieved in order to

analysis the data of Question 18 and Question 32. And the final results for two kinds of respondents are shown as Figure 5.18 and Figure 5.19.

As for software engineering practitioners, we can clearly see that the most frequently type of requirement changes that they received is “Interface style modification” (40%), which can indicate that maybe interface modification is the most frequently involved in the communication of changes and it is easiest to communicate during requirement communication process due to the success rate of achieving this type of requirement change is as high as 96%. The number of software engineering practitioners that selected “Add new functions” (24%) is followed the number of “Interface style modification” and all of the requirement changes have been successfully implemented. “Traditional function modification” (20%) has been selected by 16 software engineering practitioners, which indicates that traditional function modification is a type of requirement change that involved in requirement changes communication process, and the success rate of achieving this type of requirement change is 88%. For the rest of two options, i.e. “Field/Title/Description modification” (7%) and “Function extension” (9%), they are rarely selected by software engineering practitioners, it can be concluded that maybe these two types of requirements changes are rarely involved in requirement changes communication process, and the success rate of those two types of requirement changes are 67% and 71%.

As for end-users, the most frequently type of requirement changes that they proposed is the same as software engineering practitioners, which is “Interface style modification” (57%) and the success rate of achieving this type of requirement change is as high as 90%, so maybe this type of change is easy to communicate with software engineering practitioners. Some of them usually propose “Traditional function modification” (23%) and the success rate of achieving it is 88%. For the rest of three types of requirement changes, i.e. “Field/Title/Description modification” 100%, “Add new function” 100% and “Function extension” 100%, then we can know that maybe field and function types of requirements are easy to communicate so that there will be rarely changes appeared in the communication of requirement changes, even though they are rarely proposed by end-users, the success rate of achieving this type of requirement change is really high, which is 100%.

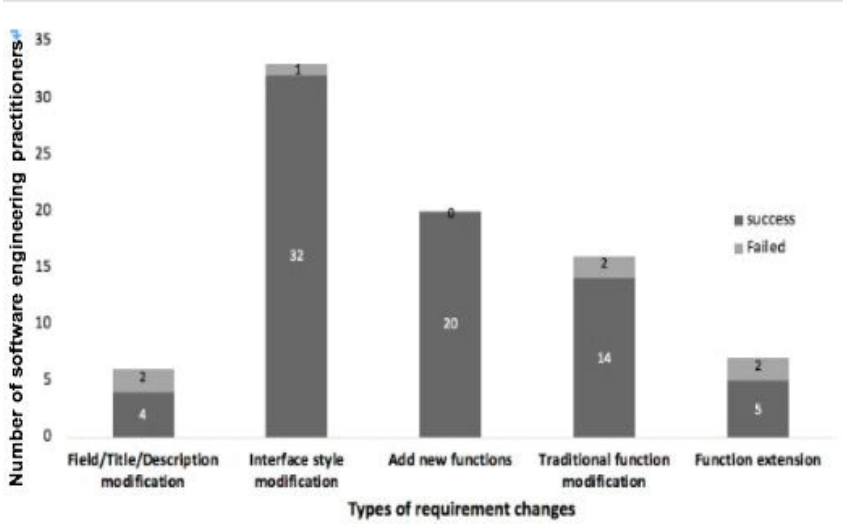

Figure 5.18: The most common type of requirements changes that software engineering practitioners met

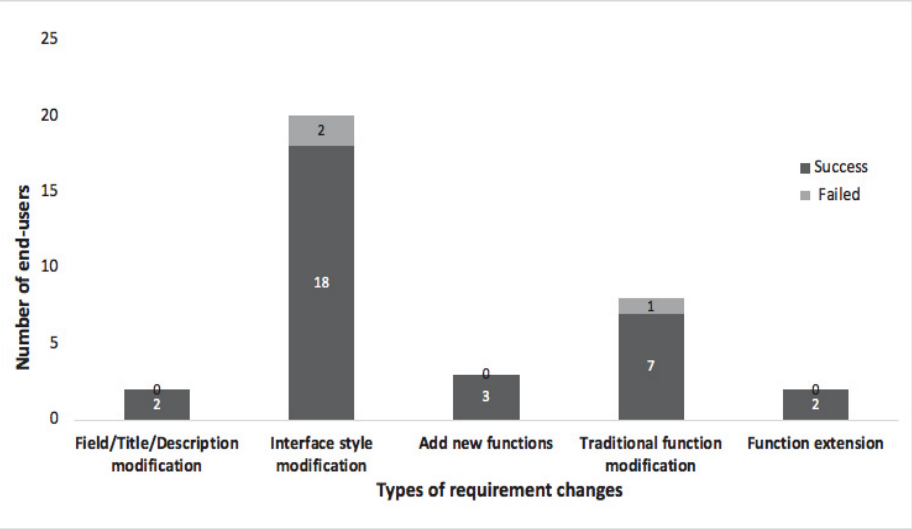

Figure 5.19: The most common type of requirements changes that end-users met

We analyzed the data of success and failure rate according to the number of success and failure requirement changes achieved by all respondents and the results are shown as Figure 5.20. We can conclude from this figure that the success rate of "Interface style modification" is highest as 94% , which means maybe this type of requirement type is easiest to communicate between software engineering practitioners and end-users. While "Add new functions" 100% and "Traditional function modification" 87.5% are relatively easy to communicate between software engineering practitioners and end-users and maybe the former is easier to communicate than the latter due to its zero-failure rate. As for the success rate of the rest of two types of requirement changes, i.e. "Field/Title/Description modification" 75% and "Function extension" 78%, we can know maybe these two types are rarely involved in the communication of requirement changes and they are a little bit difficult to communicate between software engineering practitioners and end-users.

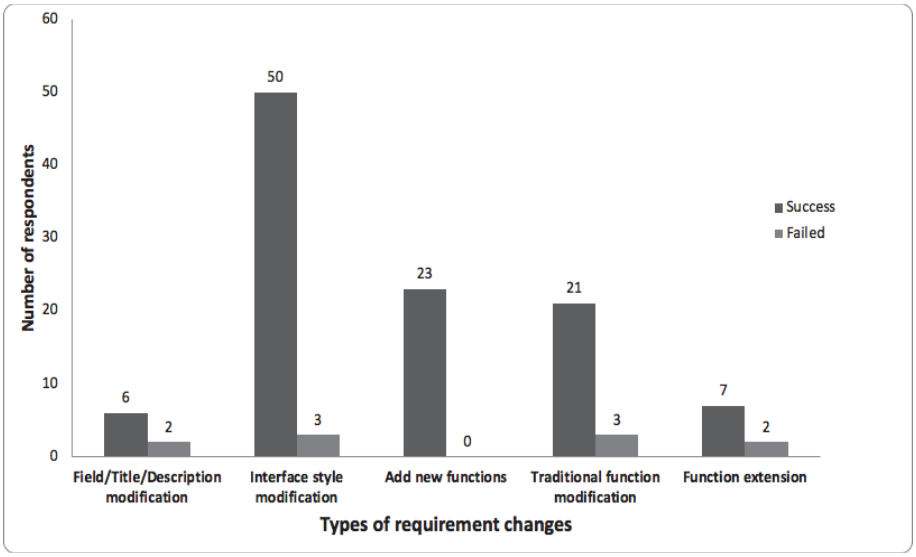

Figure 5.20: The number of successes or failures in achieving requirement changes based on different types of requirement changes

In Question 19 and Question 33, we combined and analyzed the data from software engineering practitioners and end-users. We will discuss the communication methods that different kinds of respondents prefer to use for each type of requirement change.

For "Field/Title/Description modification" type of requirement changes, the results can be shown as Figure 5.21. We found that many of software engineering practitioners prefer to use Email (39%) to communicate, because maybe it is easier to communicate with the form of text the requirement for the change of field. With the widespread popularity of agile development method, the communication method of face-to-face meeting (24%) is adopted by more and more people may be the reason of some software engineering practitioners chose this kind of communication method. For the software engineering practitioners who have rich experience may prefer to use "Writing requirement documents" (15%) because maybe they think it is more comprehensive that checking fields changes than "Video meeting" (12%) and "Voice meeting" (10%). However, due to the complex requirement documents specifications, most of them prefer to use Email rather than writing requirement documents method. As for end-users, they also don't like much about documents specifications, i.e. writing documents (3%), they prefer to use Email (51%) because they think maybe the requirement changes proposed by the form of text can be easily understood. Due to some complex of specifications, some of them may prefer to use face-to-face meeting (14%) and video meeting (23%) to communicate changes, the method that can intuitive see each other faces may decrease many misunderstandings between software engineering practitioners and end-users rather than only hear others' voice, i.e. voice meeting method (9%) thus, that's why very few respondents in both software engineering practitioners and end-users prefer to use voice meeting.

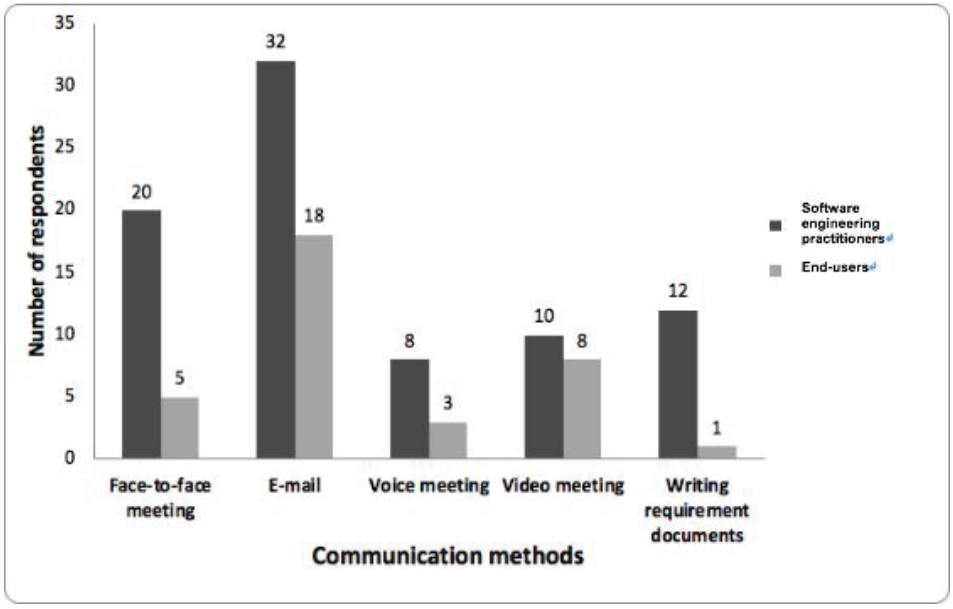

Figure 5.21: The communication method that respondents prefer to use based on the type of "Field/Title/Description modification"

For "Interface style modification" type of requirement change, the results can be shown as Figure 5.22 We found that many of software engineering practitioners prefer to use face-to-face (30%) to communicate, because maybe it is easier to communicate the change of interface by face-to-face, on the other hand, most of end-users also prefer to communicate about the modification of interface by face-to-face meeting (57%), maybe they think that face-to-face method allows the development team to be more clear about their requirements about interface. There are still many of software engineering

practitioner's chose Email (24%) because maybe some of them may more like the communication method with the form of text, then 12% of them chose writing documents. Moreover, most software engineering practitioners prefer to communicate changes within a short time period so that they will choose voice meeting (22%) rather than video meeting (11%). But for end-users, they may think that the method that can intuitive see each other faces, i.e. video meeting method (17%) may decrease many misunderstandings between software engineering practitioners and end-users rather than only hear others' voice, i.e. voice meeting method (3%). And they also prefer to use Email (17%) to communicate with the changes of interface because they can identify the interface change requirements by adding some notations in Email and don't need to consider much about the document specifications, i.e. "Writing requirement documents" (6%).

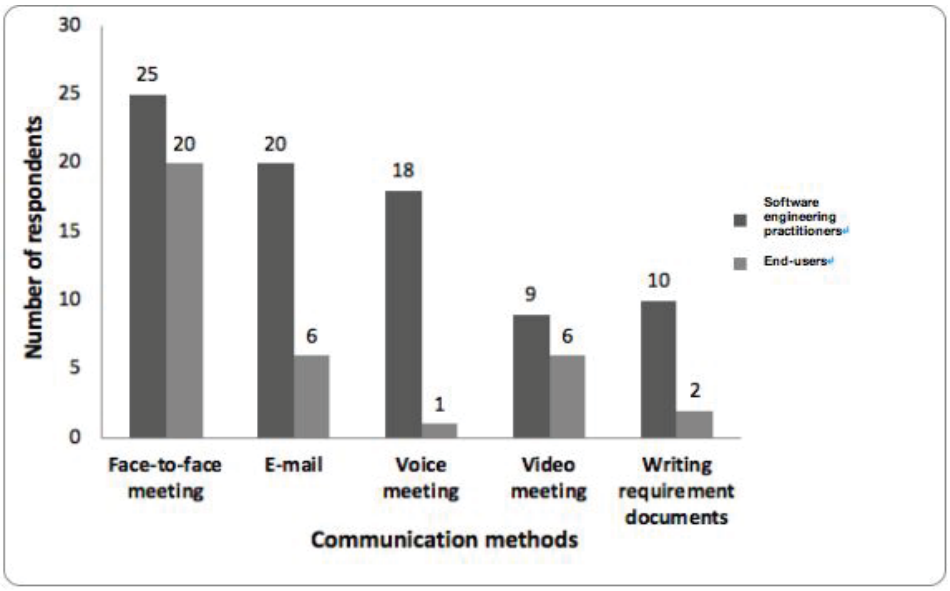

Figure 5.22: The communication method that respondents prefer to use based on the type of "Interface style modification"

For "Add new functions" type of requirement changes, the results can be shown as Figure 5.23. We found that there are still many of software engineering practitioners (24%) and end-users (29%) prefer to use face-to-face to communicate, and the percentage of "Writing requirement documents" are the same as 24%, because maybe it is important to check the description about the functions needed to add and some attributes about the new function should be described in detail in order to let the architects and developers further understand how to implement it. But some of respondents think that it is unnecessary to communicate the new function in the form of text with a such complex communication method so that Email (23%) communication method is selected by them. And some of them think that maybe it is more effective to describe a new function by using video meeting (18%) rather than voice meeting (10%). End-users also want to describe the new function needed to add explicitly so that Email (43%) method is selected by most of them, and only 2 of them chose the method with complex writing specification, i.e. writing requirement documents (6%). According to the general reason of seeing faces intuitively, 14% of them selected video meeting and only 9% of them prefer to use voice meeting.

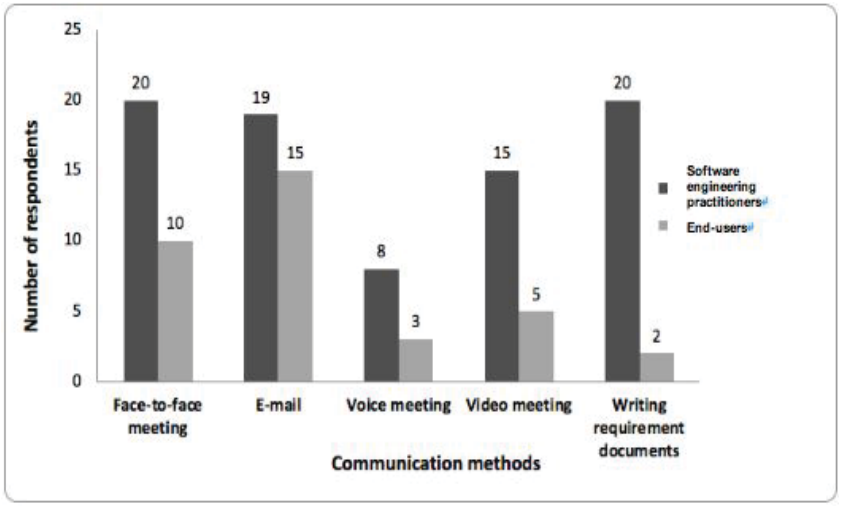

Figure 5.23: The communication method that respondents prefer to use based on the type of "Add new functions"

For "Traditional function modification" type of requirement change, the results can be shown as Figure 5.24. We found that most of software engineering practitioners prefer to use face-to-face (34%) and Email (30%) to communicate about this type of requirement change, and the percentage of "Writing requirement documents" (20%) is just follow the Email method, because maybe it is important to check the suggestions of modification about the original function in order to let the architects and developers further understand how to modify it. And the number of software engineering practitioners who prefer to use voice (6%) and video meeting (10%) are approximative. According to the general reason of seeing faces intuitively that mentioned previously, the number of end-users who prefer to use face-to-face meeting (34%) and video meeting (31%) are relatively large, some of them also prefer to use Email (26%) to communicate this type of requirement change due to the clear text form. Very few of end-users prefer to use "Writing requirement documents" (9%) and no end-user chose to use voice meeting method to communicate about this type of requirement change.

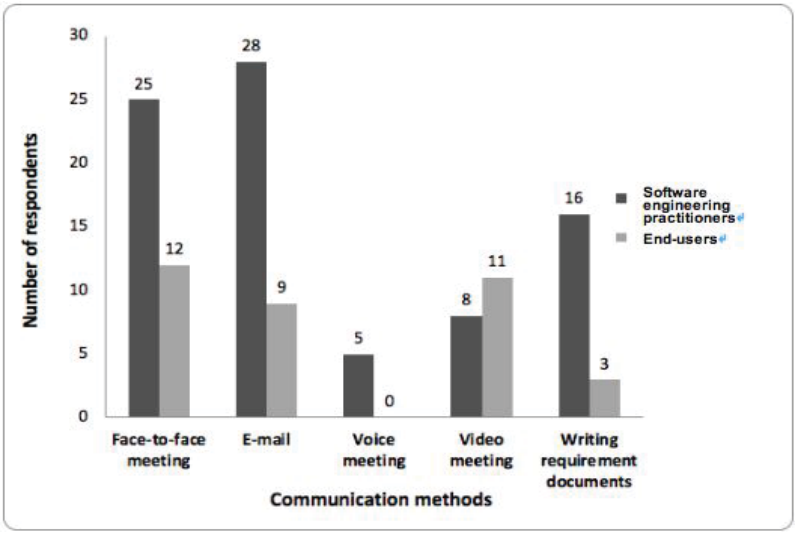

Figure 5.24: The communication method that respondents prefer to use based on the type of "Traditional function modification"

For “Function extension” type of requirement changes, the results can be shown as Figure 5.25. We found that most of software engineering practitioners prefer to use face-to-face (37%) and Email (28%) to communicate about this type of requirement change, and the percentage of “Writing requirement documents” (27%) is just follow the Email method, because maybe it is important to check the suggestions of extension about the function in order to let the architects and developers further understand how to extend it. And the number of software engineering practitioners who prefer to use video (7%) is a little larger and voice meeting (1%). According to the general reason of seeing faces intuitively that mentioned previously, the number of end-users who prefer to use face-to-face meeting (31%) and video meeting (29%) are relatively large, the number of them who prefer to use Email (29%) to communicate this type of requirement change is the same as the number that video meeting has been chosen, maybe the reason of selecting Email is the clear text form. But few of end-users prefer to use voice meeting (6%) and writing requirement documents (6%) to communicate about this type of requirement change.

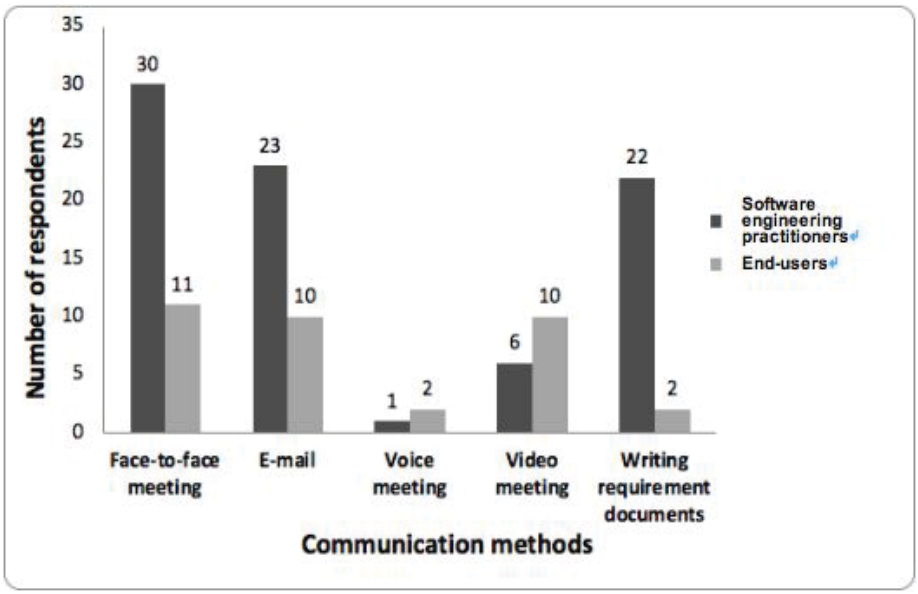

Figure 5.25: The communication method that respondents prefer to use based on the type of “Function extension”

5.5 Survey results on problems and challenges in requirement communication

In Question 20, we asked software engineering practitioners if they used technical terms when communicating requirements with users. 44% of software engineering practitioners say they have used special symbols, and the remaining 56% have never used them. In question 34, we asked the end users about the same point if there was a misunderstanding in the requirements communication because they did not understand the technical terms. What surprised us was that 75% of users said they did not have this experience, only 25% users have misunderstood due to technical terms. We present the results as a histogram in the Figure 5.26 below.

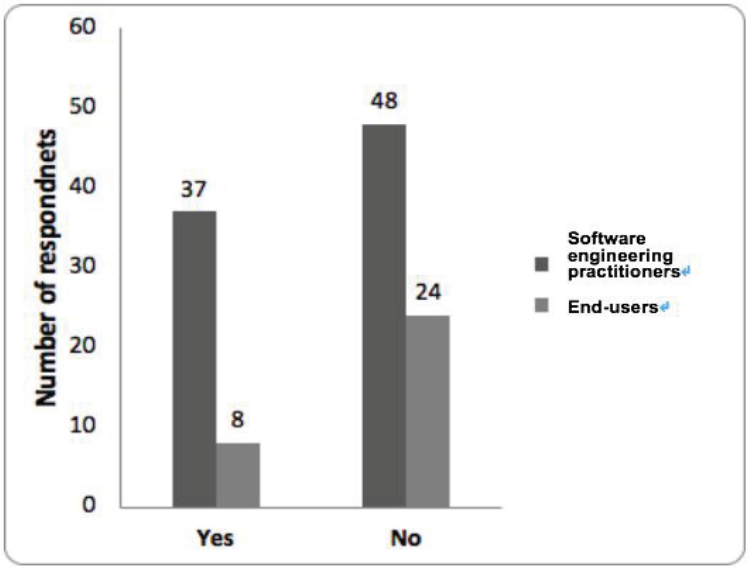

Figure 5.26 The results of the using or receiving of technical terms

It's worth mentioning that after selecting "yes", we also provided open-ended questions for participants to list the types of terms they had used or received. We received 12 answers from software engineering practitioners and 5 from end user, we summarize these technical terms.

The first is Internet terms, such as "cookies", "HTTPS". The second is hardware terms, such as "DMA", "I / O address". Software terms are most commonly mentioned by software engineering practitioners, such as "Abend", "Extensible", "Framework". Technical terms are most frequently mentioned by users, such as "cache", "CDMA", "PDU", and "Collision". Through this simple summary, we think that the use of technical terms is also a factor that causes misunderstanding in requirement communication.

For Question 21, we focused on the requirement communication problems that encountered in software engineering practitioner's' experience. According to the results in Figure 5.27, most of software engineering practitioner's encountered requirement communication problems due to the different cultures (73%). And the problems of "The actual implementation is inconsistent with the requirements document" (62%) and "Inaccurate requirement prioritization" (62%) are followed behind. We can clearly see that the most of problems encountered in requirement communication of software engineering practitioners are the three previously mentioned. What's more, almost half of software engineering practitioners also think that "Requirement changes can not be identified accurately" (43%) is the problem that they encounter relative frequently. However, a few of them encounter the problems of "Unclear requirement acquisition" (33%) and "Too much communication time lead to a long development cycle" (32%), which indicates that most of them can extract accurate requirements and won't spend too much time result in delay the development cycle.

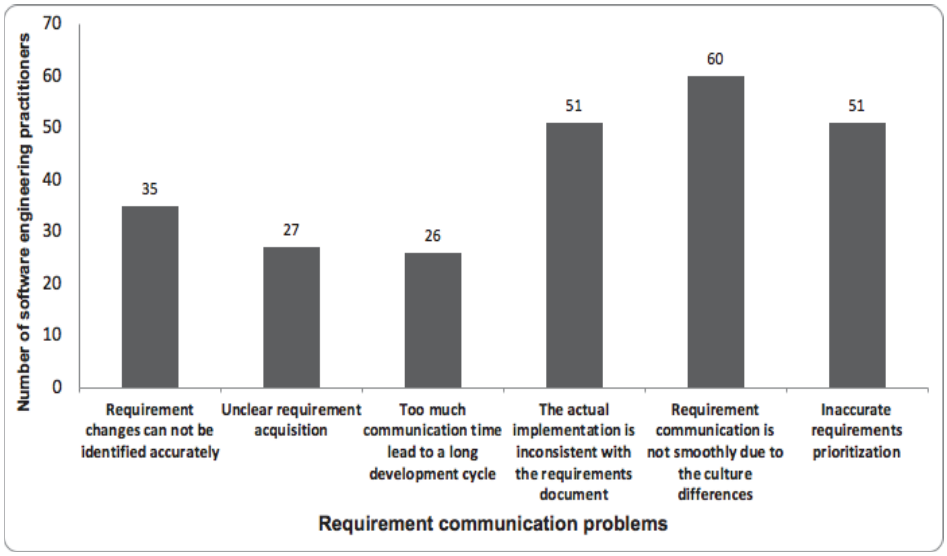

Figure 5.27: The requirement communication problems encountered in software engineering practitioner's' experience

For Question 35, we focused on the requirement communication problems that encountered in end-users' experience. According to the results in Figure 5.28, most of end-users encountered requirement communication problems due to the different cultures (86%). And the number of them who have encountered requirement communication problems of "Requirement changes can not be identified accurately" (77%) and "Too much communication time lead to a long development cycle" (71%) is almost the same, which illustrate that in the perspective of end-users, the requirements implemented by development team always have some deviations with what they originally want, and most of them think that the development cycle is still a little bit long due to the long communication time. A few of them think that the requirements extracted by development team is not really accurate (31%) during the requirement communication process. And the number of end-users encounter the problem of "Inaccurate requirement prioritized by software development team" (11%) is relatively small.

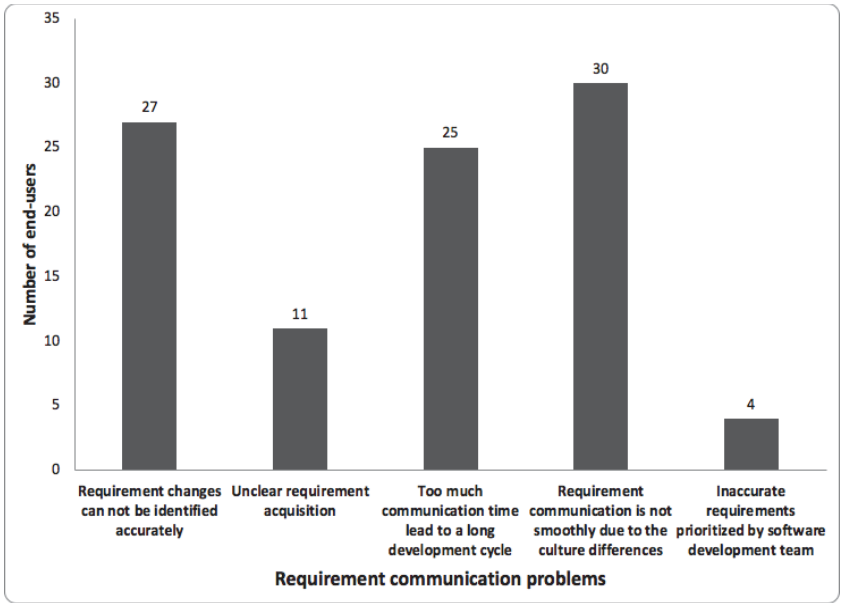

Figure 5.28: The requirement communication problems encountered in end-users' experience

Immediately following the above question about the problems encountered in the requirement communication, we set the open questions in 22 and 36 below to ask the respondents if they encountered other problems that we did not list during the requirement communication. We got three answers, two of which expressed the point that the responsibility in the requirements communication was unclear. They wrote: "When the problem occurs, all project team members are shirking responsibility, no one is responsible for this mistake, they do not acknowledge our communication information, and ultimately cause project losses." Therefore, we identified a problem: the responsible person is not clear, and false communication information is generated. Another responded that "the communicator did not understand the background of the product at all, and our communication could not be carried out smoothly". Therefore, we identified this problem as: the background of the requirement-related product is unclear, which affects the project depth and feasibility.

Comparing these answers to the challenges we have identified in the SLR, we have come up with a new challenge: manage requirement documentation.

6. Discussion

6.1 Answering the RQs

In this section, we will answer our research questions according to the analysis and results of survey and SLR (Systematic Literature Review). In addition, we will discuss the validity threats of survey and SLR part of our thesis.

6.1.1 RQ1: What are the publication trends in requirement engineering articles regarding requirement communication?

General analysis of overall publication trend

After reviewing the content of each article, we finally selected 27 articles that related to our research, we analysis the rigor and relevance of each article and category the type of study of each article and so on, all of the results are shown in Section 4.2. And from Figure 4.1, we can know that the most intensive trend of publishing of those 27 articles is between 2004 and 2015, which on average two articles published per year are about requirement communication.

From the articles we selected according to the inclusion and exclusion criteria that related to our research field, we divided them into 6 groups based on the range of years, which is shown in Table 4.6. Figure 4.4 also shows the publication trends of articles.

Based on the analysis of SLR, the answer to this question can be clearly obtained. Through SLR, we know that a preliminary study of requirement communication has been conducted since 1996 and much more researches about requirement communication conducted from 2004, the most intensive trend of the research of requirement communication is between 2004 and 2015. Although there is a small downward trend in the middle, we still believe that the overall trend is increasing. Hence, we can answer RQ1 that there are already many related paper studied about requirement communication and it is necessary to do systematic literature review to summarize their research results and get new conclusions. In other words, we can conclude that our research will contribute to the entire requirement engineering field by researching publication trend.

Analysis of the research method with the publication trend

We found that the publication trend of articles using case study as the research method has increased significantly during the publication period from 2001 to 2011. Among them, P7 and P27 use framework as research method, and P3 and P4 use the same method which is empirical study as P1 published earliest. Although P2 and P20 also adopt case study, they also combine interview with case study as a combination research method. This combined research method was also adopted by P24 that published in 2016. In the 7 articles published between 2012 and 2017, we found that compared with the articles published before 2012, the number of articles using case study as the research method decreased, but the number of articles using interview as the research method increased. In addition, among the articles on requirement communication published in the past six years, only P25 has adopted empirical study. Please see the detailed analysis in Section 4.2.1 and Figure 4.1.

Analysis of the Rigor and Relevance scores with the publication trend

We found that there is no significantly relationship between rigor and relevance scores with the publication trend, the relevance and rigor scores for each article are shown in Section 4.2.2. In Figure 4.3, we find that articles with high rigor and relevance scores are generally distributed between 2004 and 2015. In addition, we also found that articles published between 2001 and 2003 have highly rigor but low relevance scores, which indicates that early research articles on requirement communication also followed rigorous development and rules in the research process. The articles published from 2004 to 2007 and from 2012 to 2015 are highly relevant but have low rigor scores. And most articles that have both high rigor and relevance scores are distributed between 2008 and 2011, which illustrates that many articles in early research are very relevant to requirements communication field and people have found the importance of requirement communication to software development in the earlier research of requirements engineering field.

6.1.2 RQ2: How different roles in requirements engineering communicate about requirements?

RQ2.1: What are the responsibilities of the different roles, and which of them relate to requirements communication?
RQ2.2: What are the communication methods used by different roles?

We answered RQ 2.1 and RQ 2.2 based on the answers collected from the online questionnaire, the detailed statistical data are analyzed in Section 5.2, and we will answer RQ2 according to combining these two sub-questions.

As for RQ 2.1, we first identified the articles in systematic literature review that involved different roles in requirement communication, and further summarized the content and research results of identified articles. In Survey, we selected the most common roles as options in our questionnaire based on the overall introduction of stakeholders in [57], and then further judged the relationship between the choice of communication methods and roles based on the answers of the respondents, and the roles of Effectiveness of different communication methods considered the proportion of software engineering practitioner roles can be identified from the online questionnaire (Question 6). We found that most of software engineering practitioners are developers while other roles of them are Project Manager, Requirement Engineer, System Architect and Tester. In Question 7- Question 11, we formulated several responsibilities that might belong to each role and then we will identify the real responsibilities involved according to the leading votes of each role.

Responsibilities related to requirement communication extracted of software engineering practitioner

According to the numbers of answers in Question 7, we found that the number of votes for “Supervision of time and budget of developing product” and “Management of the entire team” are much more than other number of options, which can be identified the real responsibilities involved in Project Manager role while they are also related to requirement communication.

As for Requirement Engineer, we found that the number of votes for "Identification and optimization of overall project requirements", "Requirements analysis, filtration and classification", "Writing requirement documentation" are relatively similar, which can be identified the real responsibilities involved in Requirement Engineer role and "Requirements analysis, filtration and classification" is a responsibility that related to communication process. However, we believe the rest of two responsibilities which called "Requirements documents updating according to requirement changes" and "Collection of user feedback and communicating with team members" are related to communication but not much requirement engineers chose them.

As for System Architect, all of them chose the responsibilities called "Overall software architecture formulation and system decomposition vertically and horizontally" and "Coordinating and communicating with all developers" while 3 of system architects chose "Technical selection and negotiation with project manager", which means those three responsibilities are typically involved in System Architect role. What's more, "Technical selection and negotiation with project manager" and "Coordinating and communicating with all developers" are related to communication process. Although "Accurate requirements identification according to communicating with requirement engineer" is also related to communication process, it is not selected by much system architects.

As for Developer, most of software engineering practitioners are belong to this role. And as we all know that "Software programming and coding" is absolutely a typical responsibility of developers, and they also need to "Keep communicating with team members" which is related to communication process. And "Assurance of the stability and reliability of system" is also a typical responsibility of developers. However, we believe that "Assistance of testers to complete software system and module testing" is also related to communication process, there seems no much developers actually have this responsibility during their development process.

As for Tester, we can find that all of the testers have the responsibility called "Communicate about testing results with developers" while it is also related to communication process, "Conduction of system functional testing" and "Recording test results and writing test reports" are the typical responsibilities because they also have high votes. Although the responsibility called "Peer review of test cases" is related to communication process, there is no much testers actually have this responsibility during their testing process.

Responsibilities related to requirement communication extracted of end-users

As for End-user, we formulated an open question, i.e. Question 37 in order to obtain their responsibilities that involved in requirement communication process. According to their answers, we found that the mainly responsibilities regarding requirement communication process of end-users can be concluded as: provide feedback about product final delivery, provide accurate requirements to the development team and update requirement changes to the development team in time.

The communication method each role prefers to use

To answer this RQ2.2, we collect the data through asking respondents what methods they prefer to use (Question 12 & Question 26) to take the responsibilities that related to requirement communication process and what effectiveness they think to use different communication methods (Question 14 &

Question 28), then we analysis this kind of question based on different roles of respondents. The detailed data analysis is shown in Figure 5.13 and Figure 5.16.

According to Question 12, we can clearly see that most of software engineering practitioners selected face-to-face meeting and writing requirement documentations. However, the number that voted by different roles for communication methods are different. Based on the answers that combining Question 12 with Question 26, we found that both end-users and software engineering practitioners prefer face-to-face meeting much than other communication methods.

As for Project Manager, most of them also chose e-mail and writing requirement documents, which indicates that Project Manager have a relative high knowledge level for handling requirement documents and they also think it is efficient to use e-mail.

As for Requirement Engineer, we found that almost all of them chose writing requirement documents. In this case, we believe that they are familiar with requirement documents because generally they convey requirements through writing documents.

As for System Architect, they also prefer to use e-mail and writing requirement documents, because the form of text is more suitable for them to understand how to build the system environment and break down the system horizontally and vertically.

As for Developer and Tester, the voting situation is similar with System Architect.

In addition, some of software engineering practitioners also chose video meeting, the reason might be that this kind of method is more like face-to-face meeting, which can effectively eliminate misunderstanding to a large extent through seeing faces each other. However, few of software engineering practitioner's chose voice meeting because they might think it will waste time through voice chat and result in some misunderstandings (e.g. unclear speech).

As for End-user, most of them chose video meeting because they think it is an alternative method for face-to-face meeting, what's more, they don't have a such high level of expertise so that maybe they don't have that technical to write documents with some specifications and neither understand the documents written by development team. Hence, the number of e-mail and voice meeting methods are chosen are much larger than writing requirement documents.

However, we found that there is no significant relationship between the choice of communicate method and the participants' role by using Chi-Square Test to analysis those data we collected, which of the detailed statistical data analysis can also be found in Section 5.2, the difference in the number of selected communication methods may come from the subjective ideas of the respondents.

Analysis of the effectiveness of communication methods based on different roles

From the answers collected from Project Manager, Requirement Engineer, Developer, Tester and End-user (Question 14 & Question 28), we found that the most effective communication method for them is the same, which is face-to-face meeting. However, as for System Architect, they think that email communication method has the most effectiveness.

From the communication methods and the corresponding effectiveness, they chose based on different roles, we can know that the most commonly communication method used by respondents is face-to-face meeting method while few of software engineering practitioners prefer to use voice meeting method and few of end-users prefer to use writing requirement documents. On the other hand, the respondents believe that face-to-face meeting is the most effective method for requirement communication process.

6.1.3 RQ3: Does various types of requirements impact requirements communication?

The most common type of requirement that respondents have ever met

To answer this question, we analysis the data collected from Question 15 and Question 29, which is clearly shown as Figure 5.17. We found that the most common requirement type that software engineering practitioners received are "Functional requirements" and "Quality attribute requirements" while these two types of requirement are also the most frequently proposed by end-users, which indicates that the two types of requirement are always involved in requirement communication process. The detailed statistical data analyzed in Section 5.3.

Different communication methods adopted by software engineering practitioners according to different requirement types

Through the answers of "Yes" in Question 16, we found that most of software engineering practitioners will adopt different communication methods according to different requirement types. For example, 35 of software engineering practitioners prefer to use written type of communication (e.g. writing requirement documents) to deal with use case requirement because they think it will be clearer for both development team and end-users to use text form of communication method.

Different communication methods adopted by end-users according to different requirement types

Almost all of end-users selected "Yes" in Question 30. Among those answers, some of them will adopt visual type of communication (e.g. face-to-face meeting, video meeting) to propose quality attribute requirement because they think it is much easier to communicate about system quality attributes visually which is contributed to reduce misunderstanding.

Hence, through the answers collected from Question 16 and Question 30, most of respondents will adopt different communication methods according to different requirement types, thus, we believe that different types of requirement will impact the method selection during requirement communication process.

6.1.4 RQ4: What is the impact of requirement volatility on requirement communication?

RQ4.1: What is the impact of different kinds of requirement changes on requirement communication?

The most common type of requirement changes respondents has ever met and corresponding success rate

We collected the data from Question 17 and Question 31 to obtain the most common type of requirement changes. We found that the most common requirement change that respondents received and proposed are the same, which is "Interface style modification" and this type of requirement change is easiest to achieve according to Figure 5.18 and Figure 5.19 in Section 5.4.

Different communication method adopted by respondents for achieving different type of requirement changes

The detailed data analysis of communication methods adopted for each type of requirement change are shown as Figure 5.21 ~ Figure 5.25 in Section 5.4.

As the type of "Field/Title/Description modification", both software engineering practitioners and end-users prefer to use email to communicate how to achieve it because maybe written type of communication is more suitable for this type of requirement change.

As for the type of "Interface style modification", both software engineering practitioners and end-users prefer to use face-to-face meeting to communicate how to achieve it because they think that see each other faces intuitively is a good way to reduce the misunderstanding about how to modify the interface.

As for the type of "Add new functions", the number of software engineering practitioners who selected face-to-face meeting method is the same as the number of software engineering practitioners who selected writing requirement documents. Except for the intuitive communication type (i.e. face-to-face meeting) is preferred by them, the description of the details of new functions is also necessary for development team to further implement, thus, writing requirement documents method is also adopted by most of software engineering practitioners. However, because end-users are not familiar with the specification of requirement documents but they also prefer to use the form of text to communicate this kind of requirement change, then the number of end-users who selected email method becomes the biggest.

As for the type of "Traditional function modification", most of software engineering practitioner's selected email method because maybe it is important to check the suggestions of modification about the original function in order to let the architects and developers further understand how to modify it, but most of end-users selected face-to-face meeting method.

As for the type of "Function extension", as the popularity of Agile, it is general that both software engineering practitioners and end-users prefer to use face-to-face meeting.

The rate of success or failure in achieving requirement changes based on different types of requirement changes

From the detailed data analysis of Figure 5.20 in Section 5.4, we found that the success rate of requirement changes is different, it can also indicate which type of requirement change is easiest to communicate between software engineering practitioners and end-users and further be achieved.

Based on the analysis mentioned above, the types of requirement changes will not only impact the communication methods adopted by respondents, but also have different success rate.

6.1.5 RQ5: What are the challenges encountered in requirement communication and how to solve them?

Challenges identified in SLR and Survey

For answering this question, we organized nine challenges encountered in requirement communication from the literatures and the answers of the online questionnaire, which shown as Table 6.1.

Table 6.1: The list of identified challenges

Source	ID	Challenge
SLR	C1	Geographical and Culture
	C2	Organization (organizational architecture, relationship within organization)
	C3	Cognitive distance (domain knowledge, technical skill, process and organizational knowledge, priorities)
	C4	Psychological distance
	C5	Adoption of professional language
	C6	Selection of communication artifacts, tools or methods
	C7	Size and complexity of the software development
	C8	Requirement mapping in an abstract way
Survey	C9	Manage requirement documentation

From the table shown above, we can clearly see that eight challenges identified are from literatures while the rest is summarized from the answers of the open-question (Question 22 and Question 36).

The most common challenges encountered in Literatures and Survey

Through reviewing the literatures, we found that C1, C2 and C6 are the challenges identified by most articles.

Through collecting the data of the answers to the questionnaires, we summarized several challenges through the options that selected frequently by respondents. As for software engineering practitioners, we found that C1, C2 and C3 are the most frequently challenges encountered in their communication experience. But C1, C3, C5 are the most frequently challenges encountered in end-users' communication experience.

The deviation is obviously found from the differences between most frequently challenges encountered in SLR and respondents' experience. We can conclude that the challenges in practice are not consistent with the theory mentioned in the articles, which the differences of challenges between SLR and Survey are C2, C5 and C6.

Solutions identified in SLR

We summarized the solutions mentioned in literatures, which can be detailed shown as Table 4.8 in Section 4.3.5. Through analysis and summarize the solutions from SLR, we think the following types of solutions are shown as below:

1. Use models, patterns, frameworks, and other techniques
In the literature, many authors evaluate and even suggest some techniques for effective requirement communication. Many articles have pointed to models that can used during the requirement communication process, such as "FLOW model", "User needs recognition and requirements prioritization model" and "Software development models (waterfall and agile)". And there are three articles suggest patterns that can improve requirements communication, such as "Interdependent Requirements-driven communication pattern" and "Goal-oriented systems mode". In addition, we also get a "Four-dimensional framework with four different methodologies" and "Requirements-centered-social-network diagram" can also be used as a solution to improve requirements communication.

2. Suitable consideration of communication methods
There are four communication methods proposed by the literatures, which are listed as follow:
- Informal communication method
- Text-based and audio associated communication
- Face-to-face communication method
- Effective and flexible communication software

One article concluded that informal communication method is much more effective to bridge communication gaps. Two literatures stated that face-to-face communication method is a effective solution for team members to conduct requirement communication. One of the literatures found that it is more effective to use text-based and audio associated communication method for the members who have poor language skill. Some authors also asserted that the selection of flexible communication software is also a solution for increasing the effectiveness of requirement communication.

3. Other suggestions
The rest of literatures provided some suggestions that can effectively improve requirement communication, which are listed as below:

- Increasing the understanding the causes and risks in requirement communication
- Different organizational setups
- Conducting discussions within a development team related to asynchronous structures prior to synchronous negotiation meeting
- Improve the facilitation of requirements mapping and transfer it from abstract way into an understandable form

Those suggestions also point out what should people pay attention to in software engineering field in the future.

6.2 Validity threats for the SLR and the survey

As Salkind [48] said, "From a research design standpoint, the simplest way to understand threats to validity is that a hypothesis might be tested in a manner other than what the researcher had intended—a situation not to be confused with the researcher's Failure to obtain the result he or she had expected". Therefore, the assessment of threats to validity is critical to ensuring the quality of empirical research in software engineering [49]. In this part, we will discuss the external validity, internal validity, structural validity and conclusion validity.

6.2.1 External Validity

External validity refers to the situation in other areas where the research results can be extended [49]. In other words, this type of validity refers to the universality of the findings [50].

For SLR, we selected articles based on software engineering that must be relevant to requirements communication, so excluded articles may affect the promotion of our results. In addition, we conducted 5 iterations in the literature search process, and the iteration ended without the discovery of any other paper containing all relevant and important information sources. Another threat worth considering is the possibility of a paper/database inaccessible, but we used our student account to successfully gain access to all articles

In the survey, we did not place specific identity restrictions on respondents, so this reduced the threat. Another threat is the number of responses to questionnaires. We did our best to publish the questionnaire on multiple social platforms around the world, and reminded respondents to forward the questionnaire link to relevant people around them, so as to spread the questionnaire more widely in a snowball form. Due to time limitation, we received 117 valid replies within the specified time. Based on this quantity, we think it is difficult to make general generalization results.

6.2.2 Internal Validity

Internal validity is the degree to which research establishes a trustworthy cause-and-effect relationship between treatment and outcome, which depends largely on the rigor of the study's procedures and implementation [50].

For the SLR we chose, the biggest internal threat were database selection and article sifting. We used both forward and backward snowballing and sifted through multiple databases to avoid omissions and fully identify relevant articles. In the process, we will also review the articles at any time to avoid duplication. In addition, bias in study selection is also an internal threat. We were evaluated separately by two authors, and strictly implemented the internal and external criteria we set, and the final decision was made by the supervisor.

The internal threat to the survey mainly lies in the setting of questionnaire questions. We simplified the description of the questions as much as possible to improve the understandability of the language, and designed a lot of open questions to avoid affecting the answers of participants due to the lack of understanding of the questions, so that participants could complete the questionnaire more easily.

6.2.3 Conclusion Validity

Conclusion validity means that it is proved that the research operation can be repeated [49]. SLR has many conclusion threats. The first is the bias in paper choices, and the second is the subjective interpretation of data extraction. So our literature review is done by the author first, and then by the supervisor to review and make the final decision. Moreover, cross verifying among reviewers is necessary, and the supervisor's opinion will be taken into account when differences arise.

For surveys, the size of the sample is a clear threat. We tried our best to release the questionnaire more widely, and reminded respondents that they could share the questionnaire with their friends. However, due to the limitation of time and ability, we still only got a limited number of responds. In addition, the identity of respondents is also a threat. Although we do not have clear statistics on the nationality of respondents, due to the author's nationality and social circle restrictions, we think that the majority of respondents are from China, which is also a potential threat. It is worth mentioning that in order to reduce the threat, we verified the questionnaire questions before the formal investigation, so as to reduce the threat of the final conclusion as far as possible.

6.2.4 Construct Validity

Construct validity refers to the determination of correct operational measures for the concepts being studied [49], and the other types of validity we have described can be regarded as evidence of construct validity [52].
In the snowball literature review, the start set is very important, so we referred to the supervisor’s opinion and the proposal to ensure the correctness of the start set article. In order to avoid such threats as inappropriate inclusion, exclusion criteria and research questions, we had a thorough discussion with our supervisor before we started and made several revisions before we made the final decision. In the process of article screening and data extraction, the set criteria and research questions should be fully considered.

In survey research, construct validity addresses the issue of how well whatever is purported to be measured actually has been measured [53]. In the process of our survey, the setting of questionnaire was taken into account. Before releasing the questionnaire, we verified to ensure that the description of the question is rigorous and easy to understand, and the setting of the question should be reasonable and logical, so as to avoid influencing the respondents' answers by our description as far as possible.

7. Conclusion and Future Work

7.1 Conclusion

Our research starts from three aspects to investigate the factors that affect requirement communication. These three aspects are the communication process between different roles, different requirement types, and requirement volatility. We first identified how requirements are communicated between different roles, which indicates the different communication methods that different roles will use, and then summarize the typical requirement types and types of requirements changes. Next, we understand whether the types of requirements and requirement volatility of requirements will affect the selection of communication methods by software engineering practitioners and end-users. Then the challenges that involved in requirement communication process will be organized through combining the answers of open-questions and the challenges we have obtained from SLR. Eventually, we summarized some solutions that can improve requirement communication.

In this study, we combined two research methods, which are SLR and Survey. We used the snowballing method to obtain 27 articles related to our research topic through conducting 5 iterations. We summarized the publication trends of these articles and then performed a rigor / relevant quality assessment of these articles, and several points related to our research question in the articles were also organized. In addition, we performed a narrative analysis of SLR to help us understand the intervention work and increase the transparency of each article. Next, we conducted survey by publishing an online questionnaire. Through the combination of answers of open-questions and the results of SLR, we also summarized some challenges and solutions related to requirement communication.

Through SLR, we screened research articles related to the three aspects mentioned above, and we found the number of researches on requirements communication is increasing, so it can fully show that it is necessary to summarize the previous articles related to the topic of our paper through systematic literature review, and we have analyzed research content of each article and got more meaningful results, thus, the answer of RQ1 is answered explicitly. By performing step two in the narrative analysis, we have a initial understanding of different communication methods selected by different roles, different types of requirements changes, and requirements types, and we have briefly summarized several challenges related to the requirements communication process. In addition, some measures that can reduce challenges for requirement communication have been collected by us in preparation for answering research questions.

Through the survey, we published an online questionnaire. Until the deadline of the questionnaire, we received 132 questionnaires, of which 117 were valid. In the recovered questionnaire, the responsibilities to different roles are identified and it is clearly shown which different communication methods that respondents have adopted based on different roles and whether the communication methods will be affected by different types of requirements and different types of requirement changes. Therefore, the answers to RQ2, RQ3, and RQ4 can be came up based on the above survey results. Finally, through the combination of open-questions and analysis of SLR to find out the challenges are involved in requirement communication, then RQ5 can be further answered.

It turns out that most software engineering practitioners and end-users like to use such visual communication type like face-to-face method to communicate requirements. Although different roles of respondents prefer to use different communication methods, we found that there is no significant relationship between the choice of communicate method and the participants' role. According to the answers collected by Question 16 and Question 30, we found that different types of requirements affect the communication methods selected by respondents, which means most respondents will select the most suitable communication method based on the specific requirement type. As for the requirement volatility, we found that the type of requirement change will also determine the communication methods selected by software engineering practitioners and end-users to a certain extent, and requirement volatility also affect the final result of requirement communication.

As for the challenges, we first preliminary summarized 8 challenges involved in requirement communication activities from SLR. And a new challenge is summarized from the answers of open-questions of survey. We identified four challenges that most occur in respondents' experience through the answers of respondents while three challenges that most occur in literatures are identified through SLR.

As for the solutions, we summarized several types from SLR. Some literatures proposed using models, patterns, frameworks, and other techniques can significantly improve requirement communication, some literatures mentioned suitable consideration of communication methods should be considered as a solution for improving requirement communication. What's more, some specific suggestions for improving requirement communication is proposed by some literatures, which also pointed out what people should pay attention to for requirement communication.

7.2 Future Work

Future work should focus on deeper analysis of requirement communication and ways to improve the effectiveness of requirement communication.

This paper aim to studies the impact of communication roles, requirement types, and requirement volatility on requirement communication. However, due to the limitation of our ability, what we have done can only prove that these aspects have a certain impact on the choice of requirement communication methods and the results of requirement communication. Therefore, we plan to study more accurately to get the specific impact of these aspects on requirement communication.

And in the process of reading the literatures, we realized that there are many factors that can also affect requirement communication, which provides us with research space. In the future, we think the research direction can be focused on the attributes of the requirements, and the different attributes of the requirements in the communication process can be identified through literatures, and study whether the requirement attributes are also a major influencing factor of the requirement communication.

We originally planned to interview some professional software engineering practitioners, hoping to learn more about the actual situation through more open questions. This work can be carried out in the future when time and resources permit, so as to make up for the limitation of only conducting survey. Moreover, the number of responses received in the survey conducted in this paper is not enough to

generalize a broad conclusion. Besides, the nationalities of the respondents are also relatively concentrated, so more extensive surveys should be conducted in more sufficient time in the future.

In the research, we also identified and summarized some challenges in the requirement communication. We should also spend more time analyzing these challenges for further expansion. We will pay more attention to providing readers with more specific suggestions and solutions.

References

[1] Wohlin, C. (2014, May). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering* (p. 38). ACM.

[2] Plachkinova, M., Peffers, K., & Moody, G. (2015, May). Communication artifacts for requirements engineering. In *International Conference on Design Science Research in Information Systems* (pp. 104-118). Springer, Cham.

[3] Al-Rawas, A., & Easterbrook, S. (1996). Communication problems in requirements engineering: a field study.

[4] Schneider, K., Stapel, K., & Knauss, E. (2008, September). Beyond documents: visualizing informal communication. In *2008 Requirements Engineering Visualization* (pp. 31-40). IEEE.

[5] Nurmuliani, N., Zowghi, D., & Powell, S. (2004, April). Analysis of requirements volatility during software development life cycle. In *2004 Australian Software Engineering Conference. Proceedings.* (pp. 28-37). IEEE.

[6] Gorschek, T., & Wohlin, C. (2006). Requirements abstraction model. *Requirements Engineering*, *11*(1), 79-101.

[7] Bjarnason, E., Wnuk, K., & Regnell, B. (2011, August). Requirements are slipping through the gaps—A case study on causes & effects of communication gaps in large-scale software development. In *2011 IEEE 19th international requirements engineering conference* (pp. 37-46). IEEE.

[8] Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, *31*(11), 1268-1287.

[9] Damian, D., Lanubile, F., & Mallardo, T. (2008). On the need for mixed media in distributed requirements negotiations. *IEEE Transactions on Software Engineering*, *34*(1), 116-132.

[10] Calefato, F., Damian, D., & Lanubile, F. (2007, August). An empirical investigation on text-based communication in distributed requirements workshops. In *International conference on global software engineering (ICGSE 2007)* (pp. 3-11). IEEE.

[11] Stapel, K., Knauss, E., Schneider, K., & Zazworka, N. (2011, August). Flow mapping: Planning and managing communication in distributed teams. In *2011 IEEE Sixth International Conference on Global Software Engineering* (pp. 190-199). IEEE.

[12] Fink, A. (2019). *Conducting research literature reviews: From the internet to paper*. Sage publications.

[13] Baumeister, R. F., & Leary, M. R. (1997). Writing narrative literature reviews. *Review of general psychology*, *1*(3), 311-320.

[14] Bem, D. J. (1995). Writing a review article for Psychological Bulletin. *Psychological Bulletin*, *118*(2), 172.

[15] Sternberg, R. J. (1991). Editorial. Psychological Bulletin, 109, 3-4

[16] Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly*, xiii-xxiii.

[17] Singer, J., Sim, S. E., & Lethbridge, T. C. (2008). Software engineering data collection for field studies. In *Guide to Advanced Empirical Software Engineering* (pp. 9-34). Springer, London.

[18] Kitchenham, B., & Pfleeger, S. L. (2002). Principles of survey research part 4: questionnaire evaluation. *ACM SIGSOFT Software Engineering Notes*, *27*(3), 20-23.

[19] Judd, Charles and, McCleland, Gary (1989). Data Analysis. Harcourt Brace Jovanovich. ISBN 0-15-516765-0.

[20] Sandelowski, M. (2000). Combining qualitative and quantitative sampling, data collection, and analysis techniques in mixed‐method studies. *Research in nursing & health*, *23*(3), 246-255.

[21] The similarities and differences between Qualitative analysis and Quantitative analysis and their advantages and disadvantages. (N. September 28). Retrieved April 16, 2018, from weisdu: http://www.wesiedu.com/zuoye/6251989816.html

[22] Pearson, K. (1894). Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, *185*, 71-110.

[23] Rodgers, M., Sowden, A., Petticrew, M., Arai, L., Roberts, H., Britten, N., & Popay, J. (2009). Testing methodological guidance on the conduct of narrative synthesis in systematic reviews: effectiveness of interventions to promote smoke alarm ownership and function. *Evaluation*, *15*(1), 49-73.

[24] Popay, J., Roberts, H., Sowden, A., Petticrew, M., Arai, L., Rodgers, M., ... & Duffy, S. (2006). Guidance on the conduct of narrative synthesis in systematic reviews. *A product from the ESRC methods programme Version*, *1*, b92.

[25] Coughlan, J., & Macredie, R. D. (2002). Effective communication in requirements elicitation: a comparison of methodologies. *Requirements Engineering*, *7*(2), 47-60.

[26] Liskin, O. (2015, March). How artifacts support and impede requirements communication. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (pp. 132-147). Springer, Cham.

[27] Bjarnason, E., & Sharp, H. (2017). The role of distances in requirements communication: a case study. *Requirements Engineering*, *22*(1), 1-26.

[28] Fricker, S., & Glinz, M. (2010, September). Comparison of requirements hand-off, analysis, and negotiation: case study. In *2010 18th IEEE International Requirements Engineering Conference* (pp. 167-176). IEEE.

[29] Marczak, S., Damian, D., Stege, U., & Schröter, A. (2008, September). Information brokers in requirement-dependency social networks. In 2008 16th IEEE International Requirements Engineering Conference (pp. 53-62). IEEE.

[30] Niinimaki, T., Piri, A., Lassenius, C., & Paasivaara, M. (2010, August). Reflecting the choice and usage of communication tools in gsd projects with media synchronicity theory. In *2010 5th IEEE International Conference on Global Software Engineering* (pp. 3-12). IEEE.

[31] Stapel, K., Knauss, E., & Schneider, K. (2009, August). Using flow to improve communication of requirements in globally distributed software projects. In *2009 Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills* (pp. 5-14). IEEE.

[32] Niinimaki, T., Piri, A., & Lassenius, C. (2009, July). Factors affecting audio and text-based communication media choice in global software development projects. In *2009 Fourth IEEE International Conference on Global Software Engineering* (pp. 153-162). IEEE.

[33] Gallivan, M. J., & Keil, M. (2003). The user–developer communication process: a critical case study. *Information Systems Journal*, *13*(1), 37-68.

[34] Abelein, U., & Paech, B. (2012, June). A proposal for enhancing user-developer communication in large IT projects. In *2012 5th International Workshop on Co-operative and Human Aspects of Software Engineering (CHASE)* (pp. 1-3). IEEE.

[35] Abelein, U., & Paech, B. (2014, April). State of practice of user-developer communication in large-scale it projects. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (pp. 95-111). Springer, Cham.

[36] Marczak, S., & Damian, D. (2011, August). How interaction between roles shapes the communication structure in requirements-driven collaboration. In *2011 IEEE 19th International Requirements Engineering Conference* (pp. 47-56). IEEE.

[37] Fricker, S., Gorschek, T., Byman, C., & Schmidle, A. (2010). Handshaking with implementation proposals: Negotiating requirements understanding. *IEEE software*, *27*(2), 72-80.

[38] Lang, M., & Duggan, J. (2001). A tool to support collaborative software requirements management. *Requirements Engineering*, *6*(3), 161-172.

[39] Korkala, M., Abrahamsson, P., & Kyllonen, P. (2006, July). A case study on the impact of customer communication on defects in agile software development. In *AGILE 2006 (AGILE'06)* (pp. 11-pp). IEEE.

[40] Cruzes, D. S., Moe, N. B., & Dybå, T. (2016, August). Communication between developers and testers in distributed continuous agile testing. In *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)* (pp. 59-68). IEEE.

[41] Rodina, A., Amjed, T., & Zarinah, M. K. (2012, March). An empirical assessment of the use of different communication modes for requirement elicitation and negotiation using students as a subject. In *2012 IEEE Symposium on Computers & Informatics (ISCI)* (pp. 70-74). IEEE.

[42] Fricker, S., Gorschek, T., & Glinz, M. (2008, September). Goal-oriented requirements communication in new product development. In *2008 Second International Workshop on Software Product Management* (pp. 27-34). IEEE.

[43] Kwan, I., Damian, D., & Storey, M. A. (2006, September). Visualizing a requirements-centred social network to maintain awareness within development teams. In *2006 First International Workshop on Requirements Engineering Visualization (REV'06-RE'06 Workshop)* (pp. 7-7). IEEE.

[44] Runeson, P., Host, M., Rainer, A., & Regnell, B. (2012). *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons.

[45] Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements engineering*, *11*(1), 102-107.

[46] Ivarsson, M., & Gorschek, T. (2011). A method for evaluating rigor and industrial relevance of technology evaluations. *Empirical Software Engineering*, *16*(3), 365-395.

[47] Kitchenham, B., & Pfleeger, S. L. (2002). Principles of survey research part 4: questionnaire evaluation. *ACM SIGSOFT Software Engineering Notes*, *27*(3), 20-23.

[48] Salkind, N. J. (Ed.). (2010). *Encyclopedia of research design* (Vol. 2). Sage.

[49] Zhou, X., Jin, Y., Zhang, H., Li, S., & Huang, X. (2016, December). A map of threats to validity of systematic literature reviews in software engineering. In *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)* (pp. 153-160). IEEE.

[50]Cuncic, A. (2019). Understanding Internal and External Validity. [online] Verywell Mind. Available at: https://www.verywellmind.com/internal-and-external-validity-4584479

[51] Muhammad, N. (2007). Suitability of the Requirements Abstraction Model (RAM) Requirements for High Level System Testing.

[52] Scribbr. (2019). The 4 Types of Validity | Explained with Easy Examples. [online] Available at: https://www.scribbr.com/methodology/types-of-validity

[53] Glaser, P. (2008). Response Rates. Encyclopedia of Survey Research Methods.

[54] Cooper, H. M. (2003). Editorial. Psychological Bulletin, 129, 3-9

[55] Diener-West, M. (2008). Use of the chi-square statistic. *Baltimore, Maryland: School of*.

[56] Keele, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Vol. 5). Technical report, Ver. 2.3 EBSE Technical Report. EBSE.

[57] Lauesen, S. (2002). Software requirements: styles and techniques. Pearson Education. (pp.350-351)

Appendix A

Rigor and Relevance scores

Study ID	Rigor			Rigor sum	Relevance				Relevance Sum
	Context	Study Design	Validity		User/Subject	Scale	RM	Context	
P1	0.5	0.5	0	1	1	1	1	1	4
P2	1	1	1	3	1	1	1	1	4
P3	0.5	1	1	2.5	0	0	1	0	1
P4	1	1	1	3	0	0	1	0	1
P5	1	0	0	1	0	1	0	0	1
P6	1	0.5	1	2.5	0	1	0	1	2
P7	1	0.5	0.5	2	0	1	0	0	1
P8	1	0.5	0	1.5	0	1	1	1	3
P9	1	0.5	0	1.5	1	1	1	1	4
P10	0.5	1	1	2.5	1	1	1	1	4
P11	1	1	1	3	0	1	1	1	3
P12	0.5	0	0.5	1	0	1	1	1	3
P13	1	0.5	1	2.5	1	1	1	1	4
P14	1	1	0	2	1	1	1	1	4
P15	1	0.5	0	1.5	1	1	1	1	4
P16	1	0.5	0	1.5	1	1	1	1	4
P17	1	0	0	1	0	0	1	1	2
P18	1	0.5	0	1.5	0	1	1	1	3
P19	1	0.5	1	2.5	1	1	1	1	4
P20	1	0.5	1	2.5	1	1	1	1	4
P21	0.5	0.5	0	1	1	1	1	1	4
P22	0.5	0	0	0.5	0	1	1	1	3
P23	1	0.5	0	1.5	1	1	1	1	4
P24	1	0.5	0.5	2	1	1	1	1	4
P25	0.5	0.5	1	2	0	1	1	1	3
P26	1	0	0.5	1.5	1	1	1	0	3
P27	0.5	0	0	0.5	0	1	1	0	2

Appendix B

Online questionnaire in English

A survey on the factors that affect requirement communication: regarding to your work experience

Hi! We are the master students of software engineering in BTH.
In order to research the factors affecting requirement communication between different roles, we created this questionnaire.
We are really appreciate for your participation in answering the questions.
All questionnaire data will only be used in our research and won't be leaked to any other organizations.
Thank you!

1. What role do you play in the software projects you have recently participated in? [Single Choice] *

○Software engineering practitioner (Jump to Question 2)

○End user (Jump to Question 23)

○None of them (Jump to Question 38)

2. Which of the following company fields are you belong to? [Single Choice] *

○Transportation

○Medical and health

○Human Resources

○Bank

○Self-media

○Cultural education

○E-commerce

○IT / software development

○Other

3. Your knowledge in the Requirement Engineering field? [Single Choice] *

○Primary

○Intermediate

○Expert

4. How much work experience do you have in the current field? [Single Choice] *

○Less than 1 year

○1-3 years

○4-6 years

○More than 6 years

5. What is the size of your team? [Single Choice] *

○0-9 team members

○10-20 team members

○20-30 team members

○More than 30 team members

6. Which of the following is your current position? [Single Choice] *

○Project manager (Jump to Question 7)

○Requirement engineer (Jump to Question 8)

○System architect (Jump to Question 9)

○Developer (Jump to Question 10)

○Tester (Jump to Question 11)

7. What are your responsibilities of your role, i.e. *project manager* in requirement engineering? [Multiple choices] * (Jump to Question 12 after answering this question)

□Market research conduction and requirements mining

□Product planning and designing in the early stage

□Management of the entire team

□Supervision of time and budget of developing product

□Product spreading and promotion

8. What are your responsibilities of your role, i.e. *requirement engineer* in requirement engineering? [Multiple choices] * (Jump to Question 12 after answering this question)

□Identification and optimization of overall project requirements

□Requirements analysis, filtration and classification

□Writing requirement documentation

□Requirements documents updating according to requirement changes

□Collection of user feedback and communicating with team members

9. What are your responsibilities of your role, i.e. *system architect* in requirement engineering? [Multiple choices] * (Jump to Question 12 after answering this question)

□Accurate requirements identification according to communicating with requirement engineer

□System feasibility verification and technical risks assessment

□Overall software architecture formulation and system decomposition vertically and horizontally

□Technical selection and negotiation with project manager

□Coordinating and communicating with all developers

10. What are your responsibilities of your role, i.e. *developer* in requirement engineering? [Multiple choices] * (Jump to Question 12 after answering this question)

□Software programming and coding

□Writing related technical documents

□Conduction of software unit tests

□Assistance of testers to complete software system and module testing

□Functional designing and interface beautification

□Assurance of the stability and reliability of system

□Keep communicating with team members

11. What are your responsibilities of your role, i.e. *tester* in requirement engineering? [Multiple choices] * (Jump to Question 12 after answering this question)

□Writing test cases according to the content of the project requirements

□Test environment establishment

□Conduction of system functional testing

□Recording test results and writing test reports

□System defects tracking and analysis

□Peer review of test cases

□Communicate about testing results with developers

12. In your current team mode, which of the following method that you prefer to conduct requirement communication? [Single choice] *

○Face-to-face meeting

○E-mail

○Voice meeting

○Video meeting

○Writing requirement documents

13. Do you have any other ways of requirement communication we didn't mention above? [Single choice] *

○Yes
Please identify the communication methods you used.

○No

14. Please choose the effectiveness of the following requirement communication methods based on your work experience. [Matrix single choice] *

	Not at all	Slightly effective	Effective	Extremely effective	No idea
Face-to-face meeting	○	○	○	○	○
E-mail	○	○	○	○	○
Voice meeting	○	○	○	○	○
Video meeting	○	○	○	○	○
Writing requirement documents	○	○	○	○	○

15. What is the most common type of requirements you have received in your experience of requirement communication? [Single choice] *

○Use case requirements

○Functional requirements

○Quality attribute requirements

○Data requirements

16. Do you adopt different requirements communication methods for different types of requirements? [Single choice] *

○Yes _____
Please give an example of what requirement methods you have adopted when you receive different types of requirement.

○No

17. What is the most common type of requirement changes you have received in your experience of requirement communication? [Single choice] *

○Field/Title/Description modification

○Interface style modification

○Add new functions

○Traditional function modification

○Function extension

18. Was the most common type of requirement changes you received successfully implemented or not? [Single choice] *

○Yes

○No

19. For different types of requirements change, which communication method do you prefer to communicate within your team? [Matrix single choice] *

	Face-to-face meeting	E-mail	Voice meeting	Video meeting	Writing requirement documents
Field/Title/Description modification	○	○	○	○	○
Interface style modification	○	○	○	○	○
Add new functions	○	○	○	○	○
Traditional function modification	○	○	○	○	○
Function extension	○	○	○	○	○

20. Have you ever used professional terms (e.g. notations, models) to communicate with end-users ? [Single choice] *

○Yes
Please identify the specific professional terms you have used to communicate with end-users

○No

21. What are the problems of requirement communication encountered in your experience? [Multiple choices] *

□Requirement changes can not be identified accurately

□Unclear requirement acquisition

□Too much communication time lead to a long development cycle

□The actual implementation is inconsistent with the requirements document

□Requirement communication is not smoothly due to the culture differences

□Inaccurate requirements prioritization

22. Do you have any other problems of requirement communication encountered in your experience? [Single choice] * **(Jump to Question 38 after answering this question)**

○Yes
Please identify the requirement communication problems.

○No

23. Which of the following company fields are you belong to? [Single choice] *

○Transportation

○Medical and health

○Human resource

○Bank

○Self-media

○Cultural education

○E-commerce

○IT / software development

○Other

24. Your knowledge in the Requirement Engineering field? [Single choice] *

○Primary

○Intermediate

○Expert

25. How much experience do you have on communicating with professional software teams? [Single choice] *

○Less than 1 year

○1-3 years

○4-6 years

○More than 6 years

26. Which of the following method that you prefer to conduct requirement communication? [Single choice] *

○Face-to-face meeting

○E-mail

○Voice meeting

○Video meeting

○Writing requirement documents

27. Do you have any other ways of requirement communication we didn't mention above? [Single choice] *

○Yes
Please propose some examples that we didn't mention above.

○No

28. Please choose the effectiveness of the following requirement communication methods based on your work experience. [Matrix single choice] *

	Not at all	Slightly effective	Effective	Extremely effective	No idea
Face-to-face meeting	○	○	○	○	○
E-mail	○	○	○	○	○
Voice meeting	○	○	○	○	○
Video meeting	○	○	○	○	○
Writing requirement documents	○	○	○	○	○

29. What is the most common type of requirements you have provided to software development team in your work experience of requirement communication? [Single choice] *

○Use case requirements

○Functional requirements

○Quality attribute requirements

○Data requirements

30. Do you adopt different requirements communication methods for different types of requirements? [Single choice] *

○Yes
Please give an example of what requirement methods you have adopted when you receive different types of requirement.

○No

31. What is the most common type of requirement changes you have proposed to the software development team in your experience of requirement communication? [Single choice] *

○Field/Title/Description modification

○Interface style modification

○Add new function

○Traditional function modification

○Function extension

32. Was the most common requirements change you received successfully implemented or not? [Single choice] *

○Yes

○No

33. For different types of requirements change, which communication method do you prefer to propose the changes to software development team? [Matrix single choice] *

	Face-to-face meeting	E-mail	Voice meeting	Video meeting	Writing requirement documents
Field/Title/Description modification	○	○	○	○	○
Interface style modification	○	○	○	○	○
Add new function	○	○	○	○	○
Traditional function modification	○	○	○	○	○
Function extension	○	○	○	○	○

34. Have you ever had a communication problem due to misunderstand the professional software development team's technical terms (e.g. notations, models)? [Single choice] *

○Yes
Please identify the professional terms you have met.

○No

35. What are the problems of requirement communication encountered in your experience? [Multiple single choices] *

□Requirement changes can not be identified accurately

□Unclear requirement acquisition

□Too much communication time lead to a long development cycle

□Requirement communication is not smoothly due to the culture differences

□Inaccurate requirements prioritized by software development team

36. Do you have any other problems of requirement communication encountered in your experience? [Single choice] *

○Yes
Please identify requirement communication problems.

○No

37. In the software development process, what responsibilities do you have as an end-user regarding requirement communication? [Fill in the blank] *

_____

38. Do you mind leaving your personal contact information and name so that we can conduct our follow-up surveys? [Fill in the blank]

First name:	_____
Last name:	_____
E-mail address:	_____
We chat account:	_____

Appendix C

Online questionnaire in Chinese

一项关于影响需求沟通因素的调查

您好，我们是布莱京理工学院的学生。为了调查影响不同角色之间需求沟通的因素，我们制定了这项问卷。我们很感谢您的参与。
所有的问卷数据都只用于此次调查，不会泄露给外部机构。谢谢!

1. 您在最近参与的软件项目中扮演什么角色? *

○软件工程从业人员（跳至问题2）
○最终用户（跳至问题23）
○没有一个（跳至问题38）

2. 您属于以下哪个公司领域? *

○交通
○医疗保健
○人力资源
○银行
○自媒体
○文化教育
○电子商务
○IT /软件开发
○其他

3. 您在需求工程领域的知识水平是? *

○初级
○中等
○专家

4. 您在当前领域有多少工作经验? *

○少于一年
○1-3 年

○4-6 年

○超过6年

5. 您的团队规模多大？ *

○0-9人

○10-20人

○20-30 人

○超过30人

6. 您目前的工作角色是什么？ *

○项目经理（跳至问题7）

○需求工程师（跳至第8个问题）

○系统架构师（跳至第9个问题）

○开发人员（跳至第10个问题）

○测试人员（跳至问题11）

7. 您的角色（即需求工程中的项目经理）的职责是什么？ [多项选择] *（回答此问题后跳至问题12）

□市场调查进行和需求挖掘

□早期的产品规划设计

□整个团队的管理

□开发产品的时间和预算的监督

□产品推广

8. 您对角色的职责是什么，即需求工程中的需求工程师？ [多项选择] *（回答此问题后跳至问题12）

□确定和优化总体项目要求

□需求分析，过滤和分类

□书面要求文件

□根据需求变更更新需求文件

□收集用户反馈并与团队成员进行沟通

9.您的角色（即需求工程中的系统架构师）的职责是什么？ [多项选择] *（回答此问题后跳至问题12）

□根据与需求工程师的沟通进行准确的需求识别

□系统可行性验证和技术风险评估

□整体软件架构的表述和系统的纵向和横向分解

□与项目经理进行技术选择和谈判

□与所有开发商进行协调和沟通

10.您对角色（即需求工程中的开发人员）的职责是什么？ [多项选择] *（回答此问题后跳至问题12）

□软件编程与编码

□撰写相关技术文件

□进行软件单元测试

□协助测试人员完成软件系统和模块测试

□功能设计与界面美化

□保证系统的稳定性和可靠性

□与团队成员保持沟通

11.您的角色，即需求工程中的测试员，您的职责是什么？ [多项选择] *（回答此问题后跳至问题12）

□根据项目要求的内容编写测试用例

□测试环境的建立

□进行系统功能测试

□记录测试结果并编写测试报告

□系统缺陷跟踪与分析

□测试用例的同行评审

□与开发人员沟通测试结果

12.在您当前的团队模式下，您希望通过以下哪种方法进行需求沟通？ [单选] *

○面对面的会议

○电子邮件

○语音会议

○视频会议

○书面要求文件

13.您还有其他我们上面没有提到的需求沟通方式吗？ [单选] *

○是_____

请确定您使用的通信方式。

○没有

14. 请根据您的工作经验选择以下需求沟通方法的有效性。 [矩阵单选] *

	一点也不	稍微有效	有效	非常有效	不知道
面对面会议	○	○	○	○	○
电子邮件	○	○	○	○	○
语音会议	○	○	○	○	○
视频会议	○	○	○	○	○
写需求文档	○	○	○	○	○

15.在需求沟通中，您收到的最常见的需求类型是什么？ [单选] *

○用例要求

○功能要求

○质量属性要求

○数据要求

16.对于不同类型的需求，您是否采用不同的需求沟通方法？ [单选] *

○是_____

请举例说明在收到不同类型的需求时您采用了哪些需求方法。

○没有

17.在需求沟通中，最常见的需求变更类型是什么？ [单选] *

○字段/标题/描述修改

○界面样式修改

○添加新功能

○传统功能修改

○功能扩展

18.您收到的最常见的需求变更类型是否已成功实施？ [单选] *

○是

○没有

19. 对于不同类型的需求变更，您更喜欢在团队内部进行哪种沟通方式？ [矩阵单选] *

	面对面会议	电子邮件	语音会议	视频会议	写需求文档
字段/标题/描述修改	○	○	○	○	○
界面样式修改	○	○	○	○	○
新增功能	○	○	○	○	○
传统功能修改	○	○	○	○	○
功能扩展	○	○	○	○	○

20.您是否曾经使用专业术语（例如符号，模型）与最终用户进行交流？ [单选] *

○是_____

请确定您用来与最终用户交流的特定专业术语

○没有

21.在您的体验中遇到需求沟通的问题是什么？ [多种选择] *

□需求变更无法准确识别

□不清楚的需求获取

□交流时间过多导致开发周期长

□实际执行与需求文件不一致

□由于文化差异，需求沟通不顺畅

□不正确的需求优先级

22.在您的经验中，您还有其他需求沟通问题吗？ [单选] *（回答此问题后跳至问题38）

○是_____

请确定需求沟通问题。

○没有

23.您属于以下哪个公司字段？ [单选] *

○交通
○医疗保健
○人力资源
○银行
○自媒体
○文化教育
○电子商务
○IT /软件开发
○其他

24.您在需求工程领域的知识？ [单选] *

○小学
○中级
○专家

25.您在与专业软件团队交流方面有多少经验？ [单选] *
○不到1年
○1-3岁
○4-6岁
○超过6年

26.您希望通过以下哪种方法进行需求沟通？ [单选] *
○面对面的会议
○电子邮件
○语音会议
○视频会议
○书面要求文件

27.您还有我们上面没有提到的其他需求沟通方式吗？ [单选] *
○是_____
请提出一些我们上面没有提到的示例。
○没有

28. 请根据您的工作经验选择以下需求沟通方法的有效性。 [矩阵单选] *

	一点也不	稍微有效	有效	非常有效	不知道
面对面会议	○	○	○	○	○
电子邮件	○	○	○	○	○
语音会议	○	○	○	○	○
视频会议	○	○	○	○	○
写需求文档	○	○	○	○	○

29.在需求沟通的工作经验中，您提供给软件开发团队的最常见的需求类型是什么？ [单选] *
○用例要求
○功能要求
○质量属性要求
○数据要求

30.对于不同类型的需求，您是否采用不同的需求沟通方法？ [单选] *
○是_____
请举例说明在收到不同类型的需求时您采用了哪些需求方法。
○没有

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