



# **What do we know about Testing practices in Software Startups?**

**Zhipeng Dong**  
**Mingyu Ren**

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

**Contact Information:**

Author(s):

Zhipeng Dong

E-mail: [zhdo15@student.bth.se](mailto:zhdo15@student.bth.se)

Mingyu Ren

E-mail: [mire15@student.bth.se](mailto:mire15@student.bth.se)

University advisor:

Dr. Ahmad Nauman Ghazi

Department of Software Engineering

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

Internet : [www.bth.se](http://www.bth.se)  
Phone : +46 455 38 50 00  
Fax : +46 455 38 50 57

# ABSTRACT

**Context.** With the rapid development of the software industry, innovative software products become the mainstream of the software market. Because software startups can use a few resources to quickly produce and publish innovative software products, more and more software startups are launched. Software testing is important to ensure the quality of product in software companies. Software testing is costly in software development, but if software testing is avoided, it could be costlier. Many different regular software companies spend up to 40-50% of development efforts on software testing [1] [2]. Compared with other regular software companies, time and money are finite and need to be allocated reasonably in software startups. Unreasonable allocation of time and money could lead to the failure of software startups. We don't know how much software startups spend for testing, and few research studies have investigated the testing practices in software startups. Therefore, we decided to conduct an exploratory study to know about the testing practices in software startups.

**Objectives.** The aim of the research is to investigate testing practices in software startups. In this study, we investigate software startups' structure and how to manage their test team. The test processes and test techniques used in software startups have been researched. And the main testing challenges in software startups have been investigated as well.

**Methods.** We mainly conducted a qualitative research for the study. We selected literature review and survey as the research method. The literature review method is used to get in-depth understanding of software testing practices in software companies. Survey is used to answer our research questions. We used interview as our data collection method. And in order to analyze data from interviews, we selected descriptive statistics method.

**Results.** A total of 13 responses were obtained through interviews from 9 software startups. We got results from 9 investigated software startups to structure and manage their test teams. We analyzed the common steps of test processes and classified the techniques they used in the 9 software startups. At last, we analyzed and listed the main testing challenges that are occurred in the 9 software startups.

**Conclusions.** The research objectives are fulfilled. The research questions have been answered. We got the conclusion based on 9 software startups. The 9 companies cannot represent all software startups, but we can know about test practices in software startups initially through the 13 interviews. We also found some differences about testing practice between 9 software startups and regular software companies. Our study is a primary research to explore testing practices in 9 software startups, we provided some data and analysis results of the 9 companies to the researchers who want to research some related area. In addition, our research could help someone who plans to set up a software company. They can use the data we collected to think about the testing practice in their own company. Then find out the best way to prevent and resolve the problem in testing.

**Keywords:** software startups, test practice, test techniques, structure, management, testing challenges.

# CONTENTS

<b>ABSTRACT .....</b>	<b>I</b>
<b>CONTENTS .....</b>	<b>II</b>
<b>1 INTRODUCTION .....</b>	<b>1</b>
<b>2 RELATED WORK .....</b>	<b>4</b>
2.1 SOFTWARE TESTING .....	4
2.1.1 <i>Software testing method</i> .....	4
2.1.2 <i>Software testing strategy</i> .....	5
2.1.3 <i>Software testing process</i> .....	6
2.2 SOFTWARE STARTUPS .....	7
2.3 THE IMPORTANCE OF SOFTWARE TESTING IN SOFTWARE INDUSTRY .....	9
<b>3 METHODOLOGY .....</b>	<b>11</b>
3.1 RESEARCH AIM AND OBJECTIVES .....	11
3.2 RESEARCH QUESTION .....	11
3.3 EXPECTED OUTCOME .....	13
3.4 LITERATURE REVIEW .....	13
3.4.1 <i>Description and Motivation</i> .....	13
3.4.2 <i>Steps of Literature Review</i> .....	14
3.5 SURVEY .....	16
3.5.1 <i>Description and Motivation</i> .....	16
3.5.2 <i>Data collection</i> .....	17
3.5.3 <i>Sampling method</i> .....	19
3.6 DATA ANALYSIS .....	20
3.6.1 <i>Quantitative data analysis method</i> .....	20
3.6.2 <i>Qualitative data analysis method</i> .....	21
<b>4 RESULTS AND ANALYSIS .....</b>	<b>22</b>
4.1 RQ1: WHAT IS THE STRUCTURE OF SOFTWARE STARTUPS, AND HOW THEY MANAGE THE TEST TEAM? .....	24
4.1.1 <i>Result and analysis of literature review</i> .....	24
4.1.2 <i>Result and analysis of survey</i> .....	25
4.2 RQ2: WHAT TEST PROCESSES AND TEST TECHNIQUES SOFTWARE STARTUPS FOLLOW TO ENSURE QUALITY OF END PRODUCTS? .....	28
4.2.1 <i>Result and analysis of literature review</i> .....	28
4.2.2 <i>Result and analysis of survey</i> .....	31
4.3 RQ3: WHAT ARE THE TESTING CHALLENGES THAT SOFTWARE STARTUPS ENCOUNTER IN INITIAL YEARS? .....	36
4.3.1 <i>Result and analysis of literature review</i> .....	36
4.3.2 <i>Result and analysis of survey</i> .....	37
<b>5 DISCUSSION AND LIMITATION .....</b>	<b>42</b>
5.1 DISCUSSION .....	42
5.2 LIMITATIONS AND VALIDITY THREATS .....	45
5.2.1 <i>Limitations</i> .....	45
5.2.2 <i>Validity Threats</i> .....	45
<b>6 CONCLUSION AND FUTURE WORK .....</b>	<b>47</b>
6.1 CONCLUSION .....	47
6.2 FUTURE WORK .....	48
<b>REFERENCES .....</b>	<b>49</b>
<b>APPENDIX A: SELECTED LITERATURES .....</b>	<b>53</b>

**APPENDIX B: INTERVIEW QUESTIONS.....56**

# 1 INTRODUCTION

It is observed that nowadays more and more software startups are launched. Compared with regular software companies, the main difference characteristics of software startups are high uncertainty and rapid evolution [3]. The common characteristics of software startups are high growth and high risk co-exist. Regular software companies are more robust, harmonious and have a better ability to face more challenges. During the literature study, we found 60 % software startups could not survive more than five years [4]. Therefore, there is a likelihood that software startups have higher risk to fail compare to regular software companies. As per the existing research literature, the main reasons for the failure of the software startups are lack of resources, immaturity, multiple influences and dynamic technologies [3] [5]. Although the rate of failure for the software startups is high, it is observed that there are still more and more software startups launched with innovative ideas and products. Through the literature study, we found that there are three main reasons for the increase of software startups developed: The first reason is innovation. The company through innovation tend to explore the opportunities which other people or companies did not find or develop. Then this company could find new markets, acquire new customers, or configure resources in new ways. The second reason is opportunistic orientation. It refers to the characteristics of the entrepreneurial activities that identify opportunities, take advantage of opportunities, develop opportunities and produce economic results, or come up good ideas quickly into reality. The third reason is the dynamic nature of business activities [3]. On the one hand, the entrepreneurial spirit of the software startups continues with the growth of enterprises and strengthen; on the other hand, there are plenty of opportunities to discover and use in a dynamic process.

With the rapid development in the software industry, innovative software startups became an increasing trend in software market. Software startups can use a few resources to quickly develop and market innovative software products [5]. Most of software startups aim to quickly generate revenue and obtain funding for development [3]. It means that the quality of the software project is not considered as one of the most critical thing. The company selected different software development methodologies to use according to the context of the company. Through our literature study, we found very few research studies have investigated the testing practices in software startups or made a structured investigation of startups [6]. And we found, some of the authors [7] [8] [9] only focus on addressing the general issues of startups, there is not much focus on development and testing process in software startups. Based on above reasons and the analysis of software startups, we think it is important to investigate the software startups. Because it could help practitioners make right decisions and avoid choices which could easily lead to business failure. Therefore, we selected software startups as our research object.

Nowadays, software product updates accelerating. The founder of the software startups takes the rapid produce the software product to meet the market demand as development goals for software startups. The product of software startups should be accepted by users. If the quality of software products is bad, the user's satisfaction would not be high. So, in order to improve the user's satisfaction, software startups have to ensure product quality. Software testing is important for software quality assurance [1]. Software testing is some activities which could be planned ahead and conducted systematically. It aims to find errors through executing the programs.

Testing can check whether the designed software product could meet the users' specification [1]. There are some errors in the software product, and these errors lead to software development in cost, progress and quality out of control. Errors are the attribute of software product, and it cannot be changed. Because software product is produced by people, all the work done by people is not perfect. If these errors and defect left behind when the software release is put into operation, it would eventually be exposed. If it happened, it is hard to correct the errors and it would cause bad consequences. So, the errors of software product should be discovered as much as possible in early stage, it would be easy to modify, and be invested less cost. The purpose of software testing is to find errors in the software and try to exclude them. It is not to prove that the program is wrong [10]. Software testing is considered as a necessary and important process for the development of the software industry. Software testing could help ensure that new software will be released can work properly. Through software testing, developers can be able to find out if there are technical problems with new development software before the software is released or delivered to customers. This would help ensure the software experience of the end user's software.

Through our literature study, we found testing is essential in software engineering. Many different regular software companies spend up to 40-50% of development efforts on software testing [1] [2]. Compared with other regular software companies, time and money are finite and need to be allocated reasonably in software startups. Unreasonable allocation of time and money could lead to the failure of software startups. And we found that most software startups failed because they run out of cash [11]. We are not sure that software startups would spend 40-50% of development efforts on software testing. Software startups have some differences compared with regular software companies. Such as, the funds of software startups are limited. We think software startups spend less development efforts on software testing than regular software companies. But few research studies have investigated the testing practices in software startups. Therefore, we decided to conduct an exploratory study to know about the testing practices in software startups.

To know about testing practice in software startups, we investigated test process, test techniques and testing challenges in the 9 software startups. We also investigated the structure of the startups and their way to manage test team. We used literature review and survey as research method, and we selected interviews to collect data from 9 different Chinese software startups. Then, based on the results of the interview, we got our research conclusion. We selected Chinese software startups for our investigation because the market of software development has high potential. Although there are some differences between Chinese software startups and other countries software startups, such as the scale of Chinese software startups is larger than European software startups, the characteristics of software startups are similar in both China and Europe. We are Chinese, and we have some friends worked in different Chinese software startups. It is easier to find interviewees.

Our study is a primary research to explore testing practices in 9 software startups, the contribution of the study is that our research could provide some data and analysis results of the 9 companies to the researchers who want to research some related area. In addition, our research could help someone who plans to set up a software company. They can use the data we collected to think about the testing practice in their own company. Then find out the best way to prevent and resolve the problem in testing. The 9 interviewed companies cannot represent all software

startups, but we can know about test practices in software startups initially through the 13 interviews.

This thesis report comprises of six chapters. Chapter 1 presents the introduction. In Chapter 2, Related work is discussed to give a description of existing research literature in this area of research. Chapter 3 discusses the research methodology used for our thesis. Chapter 4 presents the results and analysis obtained from the literature review and survey. Chapter 5 consists of discussion of both literature review and survey. Chapter 6 includes conclusions made by the authors and their idea of future work. After that, a list of references used in the document is presented. Finally, the appendix shows some specific information of the research.



## 2 RELATED WORK

### 2.1 Software Testing

Software testing is an investigation which can be able to provide stakeholders the quality of the product or service with information of empirical testing [12]. It is a process to identify the correctness, completeness, safety and other quality of the software. In the specified conditions, software testing can operate the program to find the program error, measure the quality of the software, and determine whether it can meet the design requirements of the assessment process. Software testing as a technical task aims to find some software bugs, as well as to ensure the product is fitness and useful [12] [13]. In software testing, there are different testing methods, testing processes, testing tools and so on.

#### 2.1.1 Software testing method

The software testing methods can be divided into static testing and dynamic testing based on whether run the program itself or not. Static testing aims to analyze the program without executing it, just examining and reviewing it [14]. The goal of static testing is check the correctness of the program by analyzing or checking the source code, structure, process, interface, etc. Static testing can through the analysis of the program static characteristics to identify deficiencies and suspicious places. Static test results can be used for further troubleshooting and provide guidance for test case selection. Dynamic testing refers to test the running program. It aims to check the difference between the operation results and the expected results, and analyze operational efficiency, robustness and other properties. This method consists of three parts: the construction of test cases, the implementation of procedures, analysis of the output of the program. To improve the quality of software, static testing involves verification and dynamic testing involves validation. The static testing is implicit, it may not find many bugs as dynamic testing, but it could use reviews, walkthroughs and inspections before the programs running which avoid some errors [15]. Most companies use static testing and dynamic testing together to ensure the quality of products.

According to the test design method, software testing method can be divided into white-box testing, black-box testing, and gray-box testing. In black-box testing, the tester cannot see the source code, also don't know the internal structure and implementation of the tested items [14]. It is applicable to some high-level testing, such as acceptance testing, system testing. The main test case in black-box testing is requirements specifications. The testers don't require professional knowledge of programming and implementation. In white-box testing, the testers can see the source code and the internal structures in detail. Based on the source code, they could select the inputs to exercise paths and then decide the appropriate outputs. The whole testing process in white-box testing is transparent, so it is usually applied in some low-level testing, such as unit testing, integration testing. The main test case in white-box testing is detailed design. The developers and testers should be with some knowledge about programming and implementation. The white-box testing is more difficult and complicated than black-box testing. The gray-box testing is the combination of the black-box testing and white-box testing, it is like a semi-transparent box in testers' eyes. In gray-box testing, the internal structure can be known partially. The testers don't need to have full access to the source code of the software [14]. Generally, integration testing is applied in gray-box testing. In general,

black-box testing happened before white-box testing, and at this stage few companies do white-box testing, the reason for this situation is that the white-box testing on the test engineers' skill requirements should be much higher, and another reason is the current software testing development is still in the initial stage, so the white-box testing is not ideal.

Software testing method can be also classified into manual testing and automatic testing. Most of the software project teams are combined with manual testing and automated testing. For software testers, automated testing is a challenge, and testers need to learn a lot of development knowledges. Manual testing is more suitable for some new testers, the biggest drawback of manual testing is low technical content. Testers select automated testing or manual testing to use based on the logical complexity and testability of tested program. If the test program has a good testability, we can choose automated testing, otherwise, we select the manual testing. For a software project, manual testing and automated testing are equally important, both are good method to ensure software quality [16].

Based on the purpose of software testing, software testing method can be divided into functional testing and non-functional testing. Functional testing, also known as behavioral testing, testers only need to consider the functions of test product, they don't need to consider the internal structure and source code of the entire software product. Generally, test cases are written according to the requirements, and depend on the interface and architecture of software product. The most common functional testing are unit testing and scenario testing. Non-functional testing aims to verify the property of software product, such as security, compatibility, or usability [17]. Non-functional testing can improve the quality of software products and improve users' satisfaction.

### 2.1.2 Software testing strategy

There are four strategies of software testing: unit testing, integration testing, system testing, acceptance testing [1]. The brief introduction is as follow:

- Unit testing: it aims to check and verify the smallest testable unit in the software during the first-round testing for making sure each unit is fully functional. In unit testing, a unit can be as a function, an individual program, and a procedure [18]. In general, white-box testing used in unit testing. Based on white-box testing, unit testing could find all kinds of errors that may exist in each module. The purpose of unit testing has 3 main aspects: consistency verification unit code and detailed design documents; design and implementation of the detailed design document tracking, find errors in the detailed design document; errors found during encoding. Unit testing is common before the formal testing.
- Integration testing: integration testing also known as assembly testing. It is the testing after the unit testing and based on unit testing, all modules are assembled into subsystems or systems accord to the design requirements. This level of software testing combines and tests the units, to find out the interface defects in the modules. The goal of integration testing is to use the components that are tested by the unit to construct the program structure based on the design requirements [19]. It includes functional testing and non-functional testing. The functional testing uses black-box testing to test the interface specification of the module under test. And the non-functional testing is for the reliability of modules.
- System testing: the system testing combines the software, computer hardware, peripherals, and network which have been confirmed together, then do the assembly testing and validation testing of the information system. The system

testing is the testing of the entire product system, the purpose is to verify whether the system satisfies the definition requirement specifications, and find out the contradiction of specifications and requirements, to put forward a perfect scheme. After the system testing found the problem, it is necessary to debug to find out the reason and position of the errors, and then correct it [20]. It is a black-box testing based on the overall system requirements specification, and it should cover all the components of the system. Objects include not only the software to be tested, but also the hardware, peripherals, and even some of the software, hardware, and interfaces that the software relies on. System testing is the first level testing for the complete application.

- Acceptance testing: The acceptance testing is the last test operation before the software is deployed. It is the software testing activities before the release of the software, and after the completion of unit testing, integration testing and system testing [21]. It is the last stage of the technical test, also called the delivery testing. The purpose of the acceptance testing is to ensure that the software is ready and that the end user can use it to perform the functions and tasks of the software [21].

### 2.1.3 Software testing process

Software testing is a process that needs a series of activities to act it. It should be planned and made a discipline for testing to ensure the quality of the testing process. The activities can be divided into five basic steps [14]:

#### 1. Planning and control

The purpose of testing planning is to define the test approach and objectives. In the planning process, the test cases and test environment should be completed. Also, the schedule of the testing process, including implementation, execution, and evaluation need to be specified for us. In addition, we should know about the completion and exit criteria.

Control is an activity about the comparison between actual progress and planning progress, then record the result to meet the plan and objectives effectively.

#### 2. Analysis and design

Test analysis and design are for reviewing the test cases, including customers' requirements, product risk, design specifications, interface, and architecture. Besides, this activity can be able to figure out the test condition, also design the test environment, determine the required infrastructure, tools and so on.

#### 3. Implementation and execution

The implementation and execution are the actual running process of software testing. In this activity, the system could use testing tools and conduct the levels of software testing. There are some major works of implementation and execution: test cases development and prioritization, then organize the test data; re-do the execution to know the bugs previously and fix them; test log for the execution; comparison between the actual result and the predicted result.

#### 4. Evaluating exit criteria and reporting

Exit criteria depend on the different condition of software testing, the code, testing time, the cost could influence the exit criteria. To evaluate exit criteria, we should know the maximum test cases could be executed and the minimum bugs we could reduce within the deadlines. We need to have the assessment for the possible changes and improvements of exit criteria. Also, we need to summary a test report.

#### 5. Test closure activities

Test closure is the last activities before the software delivery. The final tasks of software testing process include several works: check planned deliverables have

delivered or not; testware finalized and archived for reusing; handover the maintenance organization of testware for the software; evaluate the testing process and learn the experience for future testing work.

## 2.2 Software Startups

Software startups are those who have just created, and they do not have enough funds and resources. Software startups often have a shortage of funds, lack of talent (usually only the founder and a few core employees), business developed difficult and so on [22]. Software startups often start with an entrepreneur and a vision [23]. They can see the opportunities of market and use appropriate technologies to meet the market. Then, they would build a small executive team and provide the necessary technologies, the team build the product after it is established. The environment of software startups is dynamic, unpredictable, and even disordered, but it can force entrepreneurs to quickly find market space and get sustainable revenues.

There are many differences between software startups and regular software companies. The differences are mainly reflected in the following aspects:

1. Human resources collocation. In the software startups, the management of human resources is not based on mechanical theory. It is based on the startups' team building and organizational design.
2. Communication within the team. Software startups often take a straight internal communication.
3. Type of product [6]. Software startups mainly produce innovative software products.
4. Company size is different [6]. Not all software startups are small companies, but most software startups are small-scale company.

From our literature study, we found 60% startups could not survive more than five years [4] [3], and 75% venture capital fund used in startups also failed [3]. So, we think startups have a high risk to fail [22]. According to analysis CB Insights latest 146 failed startups, we got 20 reasons for the failure of startups [11]. Software startups is a kind of startups, so these reasons could also lead to the failure of software startups. We show these 20 reasons and the frequency of their occurrence in Table 2.1.

Table 2.1: Top 20 reasons for the failure of startups

Number of reason	Reason	The proportion
No.1	Failure to pivot	7%
No.2	Burn out	8%
No.3	Don't use network/advisors	8%
No.4	Legal challenges	8%
No.5	No financing/investor interest	8%
No.6	Bad location	9%
No.7	Lack passion	9%
No.8	Pivot gone bad	10%
No.9	Disharmony on team/investors	13%
No.10	Lose focus	13%
No.11	Product mis-timed	13%
No.12	Ignore customers	14%
No.13	Poor marketing	14%
No.14	Need/lack business model	17%
No.15	Poor product	17%
No.16	pricing/cost issues	18%

No.17	Get outcompeted	19%
No.18	Not the right team	23%
No.19	Ran out of cash	29%
No.20	No market need	42%

Although software startups have a risk to fail, it still has some software startups overcome the difficulties and succeed. The innovative software products have had a significant impact on human society and software industry. There are many frontier and widely influenced products, such as Dropbox, Instagram and Facebook are created by software startups [24].

Successful software startups have a lot of features, we summarize and analyze successful software startups, the reasons for their success can be classified into four categories. They are:

1. Provide the best product for the right market. If our products do not meet the requirements of the market, the customer could not buy our products. So, to ensure the success of our products, we should spend much time to ensure the products can meet market requirements.
2. Software startups don't ignore anything that could cause failure. For software startups, we cannot separate the responsibilities of departmental managers and supervisors. Because the processes in software startups in early stages are not stable, and different departments are responsible for overlapping things, the initial stage of the software startups don't need careful division of labor.
3. Software startups develop rapidly. The main reason for the failure of the software startups is that they "run out of cash." Why did they run out of cash? Because the development of software startups is not fast enough. So, if the software startups can develop fast, it can avoid some problems effectively, such as loss of competition, loss of customers, loss of staff, loss of passion, etc.
4. The team knows how to overcome the challenges. Software startups must have the ability to change their products and adapt to different change plans.

In the early stages of software startups, due to internal resources are limited, software startups often find business opportunities through utilizing a wide range of external resources [24]. In the process of transforming innovative ideas into commercial products, software startups often face some obstacles [23]:

1. Developers are inexperienced. The delivered software product cannot meet customer's requirements. Further defects could lead the product failed.
2. Product isn't really a product. When the product is customized for customers, there's no same version of the product from two customers.
3. Product has no owner. No one knows who has the power to determine the features and performance of the product.
4. No strategic plan for product development. The software startups' leadership doesn't have a clear product development goal, the decisions made during the product development process are temporary.
5. Product platform is unrecognized. The developers don't know the importance of technologies and components of the software product.

Software quality is not the most concerned of software startups. The software startups mainly focus on find opportunities to generate revenue and obtain funding for development. In the past, "startup" had many different meanings [3]. According to the research from different researchers and practitioners, there are 15 different themes about software startups. The themes are: Lack of resources, Highly reactive, Innovation, Uncertainty, Rapidly evolving, Time pressure, Third-party dependency,

Small team, One product, Low-experienced team, New company, Full organization, Highly risky, Not self-sustained, Little working experience.

According to the model proposed by Crowne [23], we can divide the lifecycle of software startups into four stages. They are: startup stage, stabilization stage, growth stage, mature stage. The startup stage aims to improve the concept of entrepreneurial ideas for software startups, it is until to the first sales. At this stage, the software startups could assemble a small executive team which have the necessary skills to build the product. The stabilization stage starts with the first scale, until the product to be stable enough to entrusted to new customer, and would not cause any cost on product development. The growth stage starts from a stable product development process and continues to market size, with the share and growth rate established. Finally, software startups develop into a mature organization, and product development becomes powerful and predictable. This is the lifecycle of software startups [25].

## **2.3 The importance of Software testing in Software industry**

The software crisis was once the most popular topic in software industry and even the entire computer industry [26]. To solve this crisis, software practitioners, experts and scholars have made a lot of efforts. Now, people have gradually realized that the software crisis is only a situation, the reason for the crisis are errors in the software, these errors have led to software development out of control in cost, schedule and quality [27]. The error in the software is a property of software product, and it cannot be changed. Because all the software products are produced by human, and as we know that all the work done by human could not be perfect. Therefore, in order to improve the quality of software products, we should explore how to avoid the errors generation and eliminate the errors that have been made, so that making the error density in the program as low as possible [28].

Software testing is a good process to avoid the errors generation and eliminate the errors that have been made [13]. Some statistics show that, in a typical software development project, the workload of software testing often accounted for more than 40-50% of the total software development effort [29] [2]. And in the cost of software development, the cost of using software testing accounted for 30% to 50% of the total cost [13]. The goal of software testing is to find the error of the software and exclude it, it is not to prove that the program is wrong.

Software testing is a primary means of discovering errors and defects in software industry [13]. It is the quality assurance of the software product. Software testing is a process not just testing source code. It includes testing of the whole process of software production, and includes testing of user requirements, product design, such as whether the product can meet user's requirements, whether it conforms to the user's habits. The software testing process and the entire software development process is basically parallel. The test plan is formulated in the requirements analysis phase. Other related work, including the formulation of test syllabus, the generation of test data, the selection of test tools, etc. should also be carried out before the testing phase.

Based on different perspective, there are two different test purposes [30] [2]. Firstly, from the user's perspective, users want to use testing to find out the errors and defects hidden in the software to consider whether the product can be accepted. Secondly, from the developer's perspective, developers want to use testing to check and correct the errors in the software product, so that the software product could

achieve the user's requirements.

As we all know, a high-quality software product can improve the users' satisfaction, it can also improve the market of the product. So, it is helpful for software companies' success. The capability of quality assurance for a software product can directly affect the company's development and survival. Software testing is an important part in software company, which is a significant and complex work in the software lifecycle, and it is extremely essential for the guarantee of software reliability [29]. Many well-known software companies always pay more attention to the software testing process. Many people think that the biggest contributors to software development are developers, this view cannot be denied, but software testers are also in a major position in software development. In fact, in software companies, software developers and testers are equally important. Software testing can ensure that the newly developed software product can work properly, it can reduce the economic losses caused by some errors in development. If the software product is not tested in development process, it is likely that some unknown problems arise after delivery. If a software product has a major problem after entering market, it is likely lead to economic losses, especially in some big companies [14]. For example, if the enterprise has wrong economic estimates, it could cause irreversible results. Therefore, software testing is an essential process in software development. It can not only identify the technical problems in the software but also can detect whether the software product meets the requirements of customers, which can help improve customer satisfaction [31]. Obviously, if a software company produce a software product with a bad quality, when the user using this product, there are a lot of errors and even cannot be used normally, it could have a great impact on the reputation of the software company. And no one is willing to use the software products developed by this company.

Although software testing is only a small part of software development, it also cannot be ignored. Software companies should pay more attention to the software testing and ensure the quality of software product. This is important for long-term development of software companies.

## 3 METHODOLOGY

### 3.1 Research Aim and Objectives

Our overall aim is to investigate the testing practice in software startups. The idea behind this research is to analyze the difference of testing practice between regular software companies and software startups. Through literature study in some regular software companies and an exploratory study in 9 Chinese software startups, the research can help to know about testing practice in software startups.

To achieve the aim, we set up 3 objectives:

- **Objective 1:** To preliminary know about the structure of software startups and how they manage the test team. The structure of a company involved in many aspects, the research shows the company structure by investigating the proportion of development team in software startups, and the proportion of test team in development team.
- **Objective 2:** To know about the test processes and test techniques software startups follow to ensure the quality of end products.
- **Objective 3:** To find out some testing challenges that software startups encounter in initial years.

### 3.2 Research Question

To achieve the aim of the research, three research questions are formulated based on three objectives respectively.

**RQ1:** What is the structure of software startups, and how they manage the test team? (**Objective 1**)

**Motivation:** A software company is made up of people with different background, skill, and work experience. Reasonable structure is the basic guarantee for the normal operation of the software company. Unreasonable structure could hinder the operation of software company, even lead to the failure of software business. Therefore, the structure of a software company is important.

Teamwork is common in software project, in order to resolve the contradictions and conflicts within the team, team management is essential in software company [32] [33]. All kinds of teams need to be managed, test team also need management. If a test team lack of management, it could affect the effectiveness of testing. The quality of the software product cannot be ensured. Based on these reasons, we decided to analyze the structure of software startups, and how they manage the test team.

**RQ2:** What test processes and test techniques software startups follow to ensure quality of end products? (**Objective 2**)

**Motivation:** The process of software testing includes requirements analysis, testing plan, writing test cases, executing the test, analyzing results and so on [34]. In different software companies, the software testing process may be different. In order to ensure the quality of the final product, the software company not only need a good test team management, but also need an effective testing process [34].

With the development of software testing industry, more and more software testing techniques has emerged [35]. For the software company, it is important to select right test techniques. If the company choose an inappropriate test technique, it would not only expand the spending, but also could not achieve the test goal, and lead to the failure of software product. And we are not very clear about which test techniques are suitable for software startups. Hence, it is helpful to further ensure the



quality of end products by knowing about test process and techniques.

**RQ3:** What are the testing challenges that software startups encounter in initial years? **(Objective 3)**

**Motivation:** Based on the literature study of software testing [13] [36] [2], we found that software testing has many benefits, but the implementation of testing would encounter many difficulties and challenges. These challenges are from the activities of stakeholders and not only from tools and method [37]. For example, users' requirements changed frequently, test tool limitations, communication of testers is not smooth etc. The effectiveness of testing depends on who involved in the process and the decisions they made [38]. Software testing often failed because of the existence of these challenges. If the test team can know the testing challenges in advance, they could reduce the probability of test failure. Therefore, we decide to investigate the testing challenges in software startups.

In order to solve the formulated research questions, we conducted a qualitative research for the study. We selected literature review and survey as the research method. The literature review method is used to get in-depth understanding of software testing practices in software companies. Survey is used to answer our research questions. Then, we analyzed the difference of testing practice between regular software companies and software startups. It is better to know about the testing practice in software startups. We used interview as our data collection method. And to analyze data from interviews, we selected descriptive statistics method. The research objects are 13 interviewees from 9 software startups which include 9 testers and 4 test team managers. We asked some questions to test team managers and testers. Through the analysis of the results from literature review and survey, we found the answers to the research questions. The overview of the qualitative research method is illustrated in Figure 3.1.

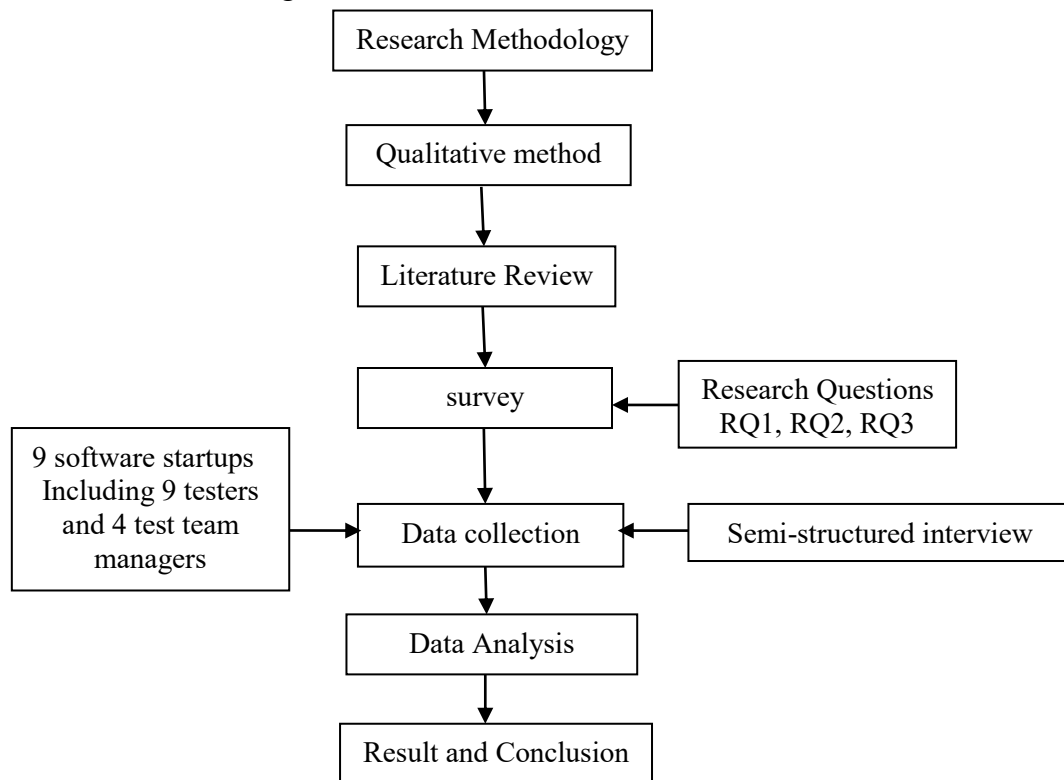


Figure 3.1: An overview of the qualitative research method

### 3.3 Expected Outcome

The following expected outcome were proposed based on the research questions:

1. The list of the percentage of developers in the 9 software startups and the percentage of testers in development team. (RQ1)
2. List the way of test team management in the 9 software startups. (RQ1)
3. The results of test techniques used in the 9 software startups. (RQ2)
4. The list of test process used in the 9 software startups. (RQ2)
5. The result of testing challenges of the 9 software startups encountered. (RQ3)

### 3.4 Literature Review

A literature review (LR) is used to search the information about the structure, management of test team, test processes, test techniques, and testing challenges in software companies. Compared with the result of three research questions from interview, it is helpful to get a better understanding of testing practice in software startups.

#### 3.4.1 Description and Motivation

To find the key concepts and theoretical framework, the literature review is selected as the qualitative research method to start the research. "A literature review is an objective, thorough summary and critical analysis of the relevant available research and non-research literature on the topic being studied" [39]. Be different from integrative review and meta-analysis, a literature review committed to providing some relevant data and source from proposed research [40], then through the analysis, reading, organizing, refining the current issues, problems or research topics of the latest progress, academic opinions or suggestions to make a comprehensive introduction and elaboration [40]. Related to the research questions, the literature review has systematic identification, location, also analysis for the literature [41]. It aims to gather and review the basis of current literature [39].

The study uses a literature review to start a qualitative research. The reason for selecting literature review as the research method is that we could determine the research focus on by comparing with the related literature. Based on the result and topic from other related research, we could rationalize the reason to do this research in the area and know about the significance of the particular research we selected. The other reason is that a literature review would gain the trust of readers. Depending on some professional research paper, the methods, conclusions, and findings mentioned in these research papers could make our research more trustworthy.

The reason for selecting literature review instead of a systematic literature review (SLR) is that SLR could take much time and the research about testing in software startups is limited. It is hard to find adequate studies for the research questions. As a secondary study, a systematic review needs more effort and well-defined methodology than traditional literature review [42]. A systematic mapping study could not be selected because it is identified and conducted based on more primary studies for future systematic reviews. The tertiary review is not chosen for the research questions since it conducts when the secondary reviews—systematic reviews exist. The tertiary review is similar to systematic literature review [42]. Hence, a literature review has been chosen as a research method for the first step of qualitative research.

### 3.4.2 Steps of Literature Review

The literature review of the research followed the step-by-step approach outlined by Galvan [43]. The proposed approach shows the step of literature review clearly and it's useful to write the review. The steps we followed for the study are as follow:

1. Decide on a topic

Selecting a review topic is the first thing to do a literature review. Since the sufficient literature is important for supporting the research, and it is difficult for the student to decide the topic [39], we should determine a clear topic. Because the research aims to investigate testing practices in software startups, very few literatures involved in both testing practices and software startups. Therefore, we expanded the scope of the study in literature review. And the topic we selected to research is about testing practices in software companies.

2. Identify the literature we'll review

After decided the topic, we searched literatures. As a traditional literature review, this step is different to systematic review. It aims to find some helpful literatures related to the topic and decide the approach from some specific information [39]. We usually search the literature resource in the computer database and electronic database. The most commonly used for retrieving information are Google Scholar, Inspec, and IEEE Xplore. To narrow the search, we selected the databases which is relevant to software engineering. It can help to determine the scope of searching quickly. So, Inspec was selected for searching literatures. From Inspec, we can search the keywords in title, abstract, author or all fields, and choose the content types we want, such as conference publications, articles, journals, eBooks and so on, to limit the search. To supplement the literatures and make the searching more comprehensive, Google Scholar was also selected for the literature review. We can select the publication year, correlation, language to find the literature in Google Scholar. In conclusion, Inspec and Google Scholar were selected for searching literatures.

During the literature review, we need to identify some appropriate keywords for searching literatures in the database. In the research, we used a literature review to investigate test practices in software companies, so the keywords should be based on this topic. First of all, the research object of literature review is software company, so the first keyword is "software company". The first research question aims to investigate the company structure and test team management. So "company structure" and "team management" can be used as the keywords. The second research question is about the test process and test techniques the software company follow to ensure the quality. So "test process" and "test techniques" can be used as the keywords. The third research question aims to find out the testing challenges in software company. So, the keyword should be "testing challenge". In summary, the keywords for searching literatures are "software company, company structure, team management, test process, test techniques, testing challenges".

The keywords we search for selecting the literature in quick search have "AND", "OR", "NOT" Boolean operators. That means we could find the literature include several keywords simultaneously, and we can search the literature which includes any of the keywords we identified, also we could find the literature exclude the keywords we identified [39]. Thus, we defined search string: (software company) AND (structure), (software) AND (team management), (software) AND (test process), (software) AND (test techniques), (software

company) AND (testing challenges), (software) AND (testing challenges). We narrowed the search after searching the literatures in the database based on the inclusion criteria and exclusion criteria defined in table 3.1.

Table 3.1: Inclusion criteria and exclusion criteria of searching literatures

Inclusion criteria	Exclusion criteria
<ul style="list-style-type: none"> <li>➤ The language of the literature must be English.</li> <li>➤ The literature can find full text online included.</li> <li>➤ The content types of the literature limited in conference paper, articles, journals, books, eBooks.</li> <li>➤ Literature related to research questions— company structure, management, testing practice, techniques, and testing challenges included.</li> <li>➤ Peer-reviewed papers included.</li> </ul>	<ul style="list-style-type: none"> <li>➤ The literatures are not written in English excluded.</li> <li>➤ Only the content or abstract are available not full text excluded.</li> <li>➤ Negative literatures excluded</li> <li>➤ The literatures are not related to software engineering excluded.</li> <li>➤ Literature not related to the research questions excluded.</li> </ul>

### 3. Analyze the literature

After identification, we analyzed and organized the selected literatures by skimming them firstly. We focus on abstract, introduction, the title of each part and conclusion [43]. We did this to exclude some worthless literatures. After skimming the literatures, we had a basic classification of the literatures. We read these articles second time in full text critically. During the analysis, we record some notes of each literature and sorted them into different categories [43]. The notes record and analyzed following information.

- The related information or data to analyze and compare with testing practice in software startups. It would be useful to review the literatures we found.
- Record some keyword and sentence as quotes to describe related literatures accurately. For example, “*Customer related Challenges of Software Testing*”, “*Software Testing Process Model*”.
- Since the related literature have the different focus, we need to record the emphases of each literature. To do the literature review, we not only record the description of the literature but also reviewed and evaluated the studies and record it.
- Record and review the author’s opinion and the research findings.
- The topic of the research is testing practices in software startups, we refined the main content of a range of literature. We organized and synthesized of what we read based on our research topic.
- Identify gaps by reading the literatures and related studies. We found few research studies have investigated the testing practices in software startups. Therefore, we decided to conduct an exploratory study to know about the testing practices in software startups.
- We found the relations of the literatures based on the categories sorted before. And used the relationships among the literatures form a framework to help write the review.
- Evaluated the references we selected. As the resource of the literature review, the references should be related to testing practices in software companies. Before writing the review, we have to make sure the reference is up to date and have a big significance of the research area.

4. Summarized the literature in table  
After identifying and analyzing the literature we selected, it is necessary to have an overall summary to show and organize the findings. We used tables to do statistics and comparative analysis. It could be able to show the results directly. We can show the keywords and concepts, research methods, the summary of research results in table [43]. We showed the table include title, authors, source, research area of the literatures we selected in Appendix A.
5. Synthesize the literature prior to writing review  
According to the analysis and summary of the literatures, we need to synthesize the literatures to prepare to write the review. First of all, we confirmed the purpose of our literature review. Use the notes we recorded during the literature analysis and organize our own understanding of the area [43]. We built a foundation of our research field. Then reassemble the recorded notes based on the categories, we organized and synthesized this literature in a new and a unique way. A literature review is not a description or annotations of many of literature, it is a new analysis or findings we created [43].  
We need to have an integral outline of the argument, we found a way to explain and justify the line of argument [43]. Then reorganized the recorded notes based on argument and the categories. Compared with the argument, there could be more literatures we need to find for our research to fill in the gaps. We had the detailed plan of writing the review. The main part of the review is the relationship of the literature and our own research, the conclusion and implications should be considered as well. These are an important basis for the literature review.
6. Writing the review  
The last and most crucial step of a literature review is writing the review. We need to clear our research topic and discuss the related literatures. We distinguished our research findings and the summary of other literature. The literature review should avoid long and confusing sentences, try to use short and accurate English to show the review concise and recapitulative [39]. It is the difference from systematic literature review, literature review doesn't need to have a clear structure, we have more freedom to write the review.

## 3.5 Survey

Survey is selected to investigate the testing practices in software startups. To answer the research questions, we conducted an exploratory study in 9 Chinese software startups. This method aims to know about structure, management of test team, test processes, test techniques, and testing challenges in the 9 software startups.

### 3.5.1 Description and Motivation

Survey is research method that aims to collect real or historical situations [45]. The purpose of survey is to collect information systematically. It often has a large research target population. And the researchers would estimate the features of the target population based on some information from a sample of respondents [46]. The survey research method has two main goals: minimize the errors of data collection and measure the errors [46]. In this method, the researchers need to describe the basic information about the sample of respondents, the target population of the research should also be provided [46]. The topic of our research is testing practices in software startups. We can provide a clear target population—testers or

test team managers in software startups. The sample of respondents are the 13 interviewees from 9 Chinese software startups. According to the answers respondents give, we could primary know about the testing practice in software startups. The survey method can achieve the research goal. It can also minimize the errors to make the statistical data more accurate. Therefore, based on the actual situation of the research and the sample of respondents, survey research methodology has been selected as a qualitative method of the study.

As we know, case study and experiment are also the common research methods in scientific research [47] [45]. A case study aims to in-depth research an individual, group or particular situation. The duration of case study is relatively long. And it often produces rich description. This research is a primary study to explore testing practice in software startups. The research target population is large. And the result of the research provided some numerical data. So, case study is not suitable for this study. Experiment research method is a kind of causal hypothesis designed, under highly controlled conditions, through the manipulation of certain factors, to test whether there is a causal link between the two phenomena [48] [49]. It is used in experimental research, and samples are relatively small. This research is an exploratory study, and the samples are large. So, the experiment method was not selected for the research. In conclusion, survey was proposed to answer the research questions.

### 3.5.2 Data collection

As we know, there are six common methods of data collection: questionnaires, interviews, focus groups, tests, observation, secondary data (e.g., personal and official documents, physical data, archived research data). And the interview is one of the most common means of data collection in survey. It can be seemed as research conversations [50]. Interviews are a way for an interviewer to ask questions, then based on the interviewees' responses, to collect objective and unbiased facts. The goal is to provide an accurate data for the research [51].

Interviews have many advantages, it can be divided into three aspects. The first is: The interview method is flexible, convenient and feasible [50]. The second is: Interview investigation is a two-way communication process. This approach has greater flexibility and it can be adjusted according to the response of the interviewees [52]. The third is: The interview method is a direct communication with the survey object, that can improve the authenticity and accuracy of the research data [50].

We know about that the questionnaire method and experiment method are also frequent used research methods. However, due to the limitations of experience and ability, we cannot ensure that the questions set up by the questionnaire could suitable for our research. And we cannot get enough responses from our survey respondents, so we did not choose questionnaire method [53]. Moreover, because this research is an exploratory study, the experimental method is also not suitable. Based on the aim, objectives and research questions, and we don't know about the current situation of software testing in software startups, interview method is the most suitable method for the study. Before we start the interview, we have to determine the type of the interview method and to determine how to do the interview.

#### 3.5.2.1 Interview questions confirmed

To confirm the interview questions, we followed the steps below:

1. Devise interview questions. We formulated some rules for devising the interview questions as follow:

- Do not ask the interviewees some broad questions [51]. Such as what is software testing?
- Do not ask the questions that are not relevant to the research. In other words, exclude redundant questions.
- Do not ask the suggestive questions. For example, most people think that software testing is important, do you think it's important?
- Do not ask meaningless or worthless questions. For example, how many toilets in your company?
- We should avoid stiff language during the interview. We need to find a proper way to speak, and try to make the interviewees willing to participate in the interviews [54].

The interview questions we formulated should not only related to the results of the literature review, but also combine with the three research questions that we put forward before. The research is to investigate testing practices in software startups, and there is a relationship between the three research questions. In order to collect as much information as possible, we decided to ask the open question to the interviewees. Each interview question is not independent, the interview process should be carried out from shallow to deep, gradually tap the nature of the research. The purpose of the interview is to know about the testing practices in the 9 software startups. It is based on the three research questions and find out the answer of research questions.

2. Review and refine the interview questions [51]. According to the interviewer's control of the interview questions, the interview can be divided into three types, they are structured interviews, unstructured interviews, and semi-structured interviews. We decided to use semi-structured interview method, because the interview questions of this method are open-ended, and the interviewers don't need to follow the formalized questions strictly. The communications of interviewees and interviewers are flexible and comfortable. It is helpful for us to get a better understanding of the research questions. Review and refine the interview questions is an important stage. At this stage, our supervisor helped us to review the interview questions. Because his experience of software testing is rich, and he can provide useful advice for us to review and refine the interview questions.
3. Update the interview questions [51]. We used semi-structured interview method, this method only has a rough basic request for the choice of interviewers, interview questions and so on [55]. Interviewers can make necessary adjustments to the interview based on the actual situation during the interview. Interviewers can adjust the way and the order of the interview questions flexibly. In addition, the time and place of the interview also can be changed flexibly by the interviewers. Because of the flexibility of semi-structured interviews, we can get different data from interviews with different interviewees [55]. After each interview, we need to relate the interview result to the interview questions. Based on the previous interview data, the existing interview questions were updated to be more efficient. The interview questions showed in appendix B.

### **3.5.2.2 Conduct the interview**

The interviews were conducted in several stages as follow:

1. Design an interview outline [56]. Before the interview, an interview outline should be designed to clarify the purpose of the interview. We list the main questions of the interview in the interview outline.

2. Introduction about the research and researcher. At the beginning of the interview, we did a self-introduction and introduced the purpose of the interview. The researcher not only need to introduce their name and occupation, but also need to describe the purpose of the research. It can help set up a more relaxed atmosphere, and ensure the full participation of the interviewees during the interviews [57] [58].
3. To ask questions properly. In order to obtain the required information, we should provide the simple, clear and accurate statement to the interviewees. The type of the interviews can be divided into open interview and closed interview. We can use these two types of interview to get as much useful information as possible. In addition, we also controlled the duration of each interviews.
4. Collection of information. The collection of information can be divided into two steps, the first step is to collect general information, the second step is to collect related information about the research [44]. In the first step, we collected the basic information about interviewees, including the name of interviewee's, their occupation, and software testing experience. The second step is to extract critical information about testing practices in software startups. In other words, it aims to find out the answer of the research questions.
5. Write an interview document. During the interview, we recorded all the information from the interviews. Interview records include audio recording and video recording. To write an interview document, we listened the records repeatedly and summarized the key information about the research. The interview document is essential to analyze the result of the research questions.

### **3.5.2.3 Interview data transcription**

We used two forms of interviews: Face-to-face interview and Skype video call interview. After each interview, we need to transcribe the interview record into text. For each interview, we used the phone to record the interview. Moreover, we created a document to record the important issues that were discussed in the interview [59]. After the interview, we transcribed the interview into a Microsoft Word document based on the audio recording of interviews. The Microsoft Word document and the audio recording of interviews are named after the name of each interviewee. We organized the documents into a folder and shared with our supervisor. In addition, to prevent the data loss, we had a backup for the file.

### **3.5.3 Sampling method**

A non-probabilistic sampling method called convenience sampling has been chosen to select interviewees. The sampling method was used for data collection, then find out the answer of research questions [60]. Sampling methods are classified into two categories: probability (Random) sampling and non-probability sampling. In probability sampling, each member of the population being selected has the same probability. It includes Simple random sampling, Systematic sampling, Stratified sampling, Multistage sampling, Multiphase sampling, Cluster sampling and so on. In non-probability sampling, the probability of each member being selected is uncertain. And the selection of the samples is nonrandom. In theory, the probability sampling is more idealized than the non-probability sampling. Because each member of population has the same probability being selected, the samples are more convictive. But sometimes we cannot use probability sampling to collect samples because of the constraints of conditions. If we want to provide some samples for the research, and



don't need to consider the probability of selection samples, we can use the non-probability sampling. Non-probability sampling is the method that the researchers select the samples based on their subjective judgment. This research is an exploratory study, and the research object is software startups, it is too numerous and scattered, it is difficult to use probability for selection, so we used non-probability sampling. This sampling method include Convenience sampling, Purposive sampling, Quota sampling, Snowball sampling [61]. In purposive sampling, the researchers select the most representative samples from the target population [62]. This sampling method requires the researchers to be familiar with the field of the study. In quota sampling, firstly, the sample groups are stratified according to certain criteria, then the samples are extracted respectively. There are many criteria for sample stratification, so the selection of standard is uncertain. In snowball sampling, the researchers select some samples randomly to investigate at first, then select the follow-up samples based on the result of the investigation. The convenience sampling is the simplest sampling method in non-probability sampling. It doesn't have too much demand for sample selection, so it could save time and money for researchers.

We selected convenience sampling for the research. Firstly, we didn't know more about the research area of software startups, so we didn't choose the purposive sampling. We are not familiar with the standard of sample stratification, so the quota sampling was excluded. And we had identified 9 software startups as the research object, so we didn't use snowball sampling. Based on these reasons above, we selected convenience sampling.

As we know, select interviewees who are familiar with the research area can ensure the reliability of the interview data [63]. So, we used convenience sampling to select 13 interviewees who have experience in software testing and working in software startups. The 13 interviewees include 4 test team managers and 9 testers. They are from 9 different Chinese software startups. In order to get useful information and data, we conducted different interviews for the 13 interviewees. And each of the interview spent 30-60 minutes. Only three interviews were face-to-face interviews. Because there was a certain distance between the interviewers and the interviewees, 10 interviews were conducted through online video.

## **3.6 Data Analysis**

The purpose of data analysis is to find the useful information that has hidden in some disorganized data, and to find out the inherent laws of the research object. In practical, data analysis can help people make judgments to take appropriate action. Data analysis is a process of collecting data, analyzing data, and getting useful information. In the research, we decided to use quantitative data analysis method and qualitative data analysis method.

### **3.6.1 Quantitative data analysis method**

Quantitative data analysis also known as quantifying research, it refers to carry out systematic empirical research for social phenomena which used statistical, mathematical or computational techniques [64]. The most important process of quantitative research is the measurement process. Quantitative data analysis research generally consists of four phases: obtaining data, data preprocessing, data analysis and generation of data analysis reports. Quantitative data are generally presented in digital percentages. Quantitative research methods often use statistical and linear programming methods to analyze data.

We used descriptive statistical analysis to collect quantitative data from

interviews. In this research, we showed the data percentage. The data percentage include two parts, the percentage of developers in software startups and the percentage of testers in development team. We also counted the number of different test methods used in these software startups, as well as the steps of the test process.

### 3.6.2 Qualitative data analysis method

Qualitative data analysis is exploratory, diagnostic and predictable method. The purpose of qualitative data analysis is not to get a precise conclusion, it is only to know about the problem comprehensively which could get a perceived understanding [65]. The main methods of qualitative research include group interviews (interview with several people), depth interviews (the answer need to be detailed), as well as a variety of projection technology. Qualitative analysis is a method for judging the nature of the object, finding out the characteristics or features of the object and knowing about the development law of the object.

First of all, we used thematic analysis method to analyze the data from the literature review and interviews. The thematic analysis simply codes the collected data, determine the subject, and generate the report. We divided the information we collected into four different categories. The first category is related to the structure of software startups. The second is related to test team management of software startups. The third category is about test processes and test techniques in software startups. The last is testing challenges in software startups. Then we used the coding to classify the data of the research. Coding is the method which can organize and classify similar data into different kinds of categories based on common characteristics, and it indicated the beginning of themes [66]. So, we used coding method for data processing of the interview. To analyze and extract the data conveniently, we used different colors to mark the data based on the importance of these data. At last, we analyzed the marked data and conclude the result.

## 4 RESULTS AND ANALYSIS

In this chapter, we show the result and analysis of the research questions based on literature review and survey. In literature review, we found few literatures have investigated the testing practices in software startups, so we expanded the scope of the research for literature review. We used literature review to investigate the testing practices in regular software companies, and used survey to know about the testing practices in the 9 software startups. We compared the difference of testing between software companies and software startups, and analyzed the results.

In literature review, we searched strings from some literature of Inspec and Google scholar. The searching obtained 638 literatures from the two databases. We narrowed the scope of these literatures based on the inclusion criteria and exclusion criteria. Then we got 58 literatures. We read the full-text of these literatures. And eventually selected 20 literatures. Table 4.1 shows the searching and selecting result.

Table 4.1: The result of searching and selecting for literature review

Database	Search string	Total literatures	Candidate literatures	Selected literatures
Inspec	(software company) AND (structure), (software) AND (team management), (software) AND (test process), (software) AND (test techniques), (software company) AND (testing challenges), (software) AND (testing challenges)	250	37	14
Google scholar	Software company structure, Software company test team management, Software test techniques used in software company, Software test process, Software testing challenges	388	21	6

The detailed information of the 20 selected literatures showed in appendix A. Most of the literatures focus on software testing. 4 of the 20 literatures are journal, 16 literatures are conference articles. The published years are from 1999 to 2017. Figure 4.1 shows the published years and figure 4.2 shows the type of literatures.

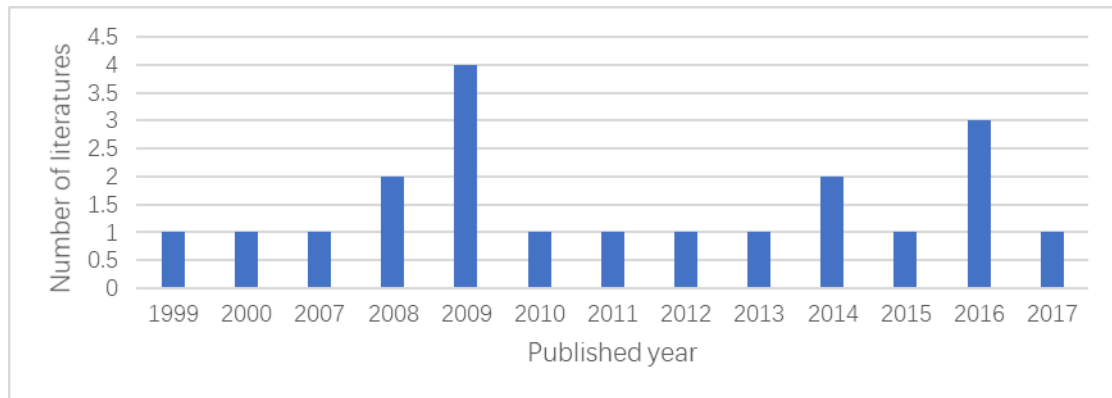


Figure 4.1: Distribution of literatures published year

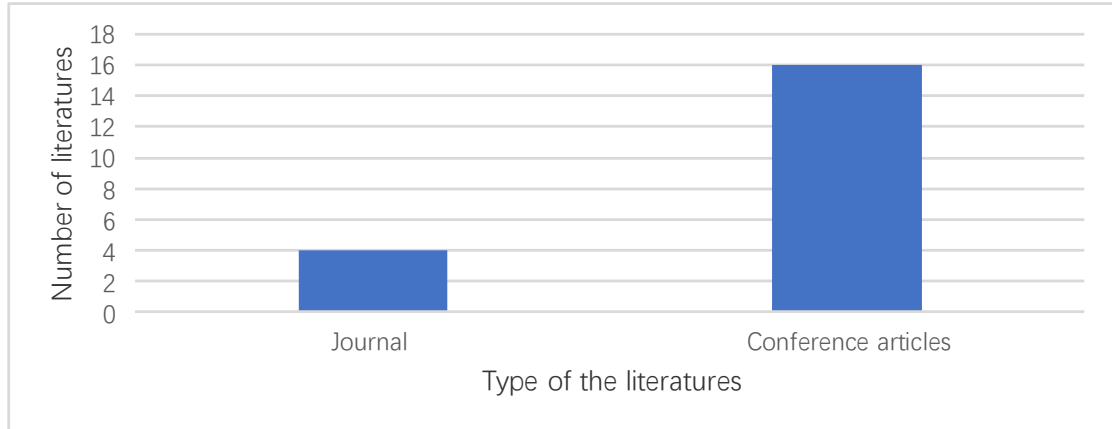


Figure 4.2: Distribution of literature types

To answer the research questions, we divided 20 selected literatures into three categories. 13 literatures focus on software test process and techniques in software company [67, 1, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78]. 4 literatures are about the structure and team management of software company [29, 79, 80, 81]. 3 literatures describe the testing challenges in software company [37, 83, 2].

In survey, the interview data was collected from the 13 interviewees in 9 software startups. The object of the interview is testers and test team managers in software startups. The interviews were based on three research questions and we collected the data then analyzed the results. Table 4.2 shows the basic information of the 9 software startups. Table 4.3 shows the basic information of the 13 interviewees in the identified startups.

Table 4.2: Basic information of 9 software startups

Company name	Company size	Development method	Type of product	Established time
ShanDong Inspur software	160	Waterfall	Cloud server	More than 1 year
Kingsoft software	150	Iterative	Phone app	2 years
ShanDong Cvices software	100	Waterfall	Phone app	More than 1 year
Hisense Group software	42	Iterative	Internet online platform	2 years
True love network	110	Spiral	Communication web	Less than 1 year
Beijing qingfeng software	65	Waterfall	Education web	1 year
Beijing Interactive streaming	150	Waterfall	Web server	More than 1 year
Panda electronics	120	Scrum	Live platform	Less than 1 year
Hangzhou Upyun software	80	Spiral	Game development	2 years

Table 4.3: Basic information of 13 interviewees

Interviewee	Position	Company name	Testing Experience
INT 1	Tester	ShanDong Inspur software	1 year
INT 2	Tester	Kingsoft software	Less than 1 year
INT 3	Tester	ShanDong Cvices software	Less than 1 year
INT 4	Tester	Hisense Group software	3 years
INT 5	Test team manager	True love network	More than 8 years
INT 6	Tester	True love network	7 years
INT 7	Test team manager	Beijing qingfeng software	3 years
INT 8	Tester	Beijing qingfeng software	1 year
INT 9	Test team manager	Beijing Interactive streaming	10 years
INT 10	Test team manager	Panda electronics	More than 5 years
INT 11	Tester	Panda electronics	3 years
INT 12	Test team manager	Hangzhou Upyun software	3 years
INT 13	Tester	Hangzhou Upyun software	1 year

As follows, we'll show the result and analysis of the three research questions from both literature review and survey.

#### 4.1 RQ1: What is the structure of software startups, and how they manage the test team?

##### 4.1.1 Result and analysis of literature review

4 of the 20 literatures are related to the structure and team management of regular software companies. Table 4.4 shows some information about the 4 literatures.

Table 4.4: Basic information of 4 literatures related to RQ1

Title of literature	Author(s)	Published year	Research area	Source	Reference
Studying the feasibility and importance of software testing: analysis	Dr.S.S.Riaz Ahamed	2009	Software testing	Journal	[29]
Overcoming barriers to self-management in software teams	Nils Brede Moe	2009	Software engineering	Journal	[79]
Choice of organizational structure for software companies in growth stage	Li jianshe	2010	Software engineering	Conference article	[80]
The influence of organizational structure on software quality: An empirical case study	Na chiappan Na gappan	2008	Software engineering	Conference article	[81]

Based on the literatures, we found most software company has complex system. In order to complete the development of software product, it often related to the interaction of different people, process and techniques. The software

development could involve hundreds of people or even thousands of people, so the organizational structure is important for software company. Moreover, we found that product quality is greatly influenced by the structure of software company. The structure of a software company can be described based on the proportion of engineer to the company, the proportion of manager to the company, and so on [81]. So, we think the proportion of the staff to the software company can help researcher simply know about the structure of the company.

We also found that the percentage of developer and tester are different in different software companies. For example, the ratio of testers and developers in Microsoft is generally 1:1, and in Windows 2000 development team, there are 1,800 testers, 900 developers, the ratio of testers and developers is 2: 1. The reason for the percentage difference is the definition of the work scope of testers and developers. In Microsoft, the unit test is doing by the tester, the equivalent of SDET (Software Development Engineer in Test) write a code to test product code of the developers, the workload is not low than developers. In Windows 2000, the products are relatively complex operating system or server software, naturally need to test a lot. The workload of testers is higher than developers.

Based on the findings, we planned to investigate the proportion of developer and testers in software startups, to know about the structure of the 9 companies.

According to the investigation of team management in software company, we found there are two ways to manage team: traditional command-and-control management, team self-management. Although team self-management has many advantages, there are also many challenges, and not all kinds of software companies are suitable for team self-management [79]. With the establishment and development of a software company, there are generally six kinds of team management based on the lifecycle of the company. The first is Line organization, the way of management is top-down, it is a leadership management of the entire company. The second is Functions organizational structure, the company is divided into multiple functional departments, each department is leadership management. The third is Line-functions organizational structure, the leader of the company can manage the department managers. The fourth is Division organizational structure, the company are divided into several project team. The project manager is responsible for the project team. The fifth is Matrix organization, the company has both project team and functional team. The last is Team-based organizational structure, the leader of the company can manage different project managers, each project team also includes some small project teams [80]. In current software development, the test team may be separated from the development team, testers and developers may not in the same project team [29]. Matrix organization and team-based organizational structure are common in some large software companies. These two types of structures have a clear division of labor and well-developed management with strictly hierarchical structure. The top department has absolute authority of the whole organization, so that it is benefit for decision-making and could save much time.

Therefore, we can know about the test team management by investigating the testers were assigned into project team or functional team. If the tester was assigned in project team, he was managed by project manager. If the tester was assigned in functional team, he was managed by test team manager.

#### 4.1.2 Result and analysis of survey

We selected 20 software startups for interview at start. However, there are 11 software startups indicated that they do not have separate testers, testing was done by the developers. So, we think some software startups didn't pay much attention to

software testing. Because there are no separate testers in the 11 software startups, they cannot give the answer of the interview question. Therefore, we did the detailed interview for 9 software startups which have separate testers.

We organized the interview results from 13 interviewees in the 9 software startups. And we found out related information about the structure of the startups and the test team management for RQ1. We marked important information with red color in Figure 4.3. Then based on the result of interview, we statistics the number of developers, testers and test team managers in the 9 software startups, we calculated the percentage of developers in the startups and the testers in development team, and show in Table 4.5. The percentage of test team manager in test team shows in Table 4.6.

- Interviewee 1: There are about 50 developers in the company. A test team have 6 people, 1 team manager and 5 testers. They have 6 development team, each development team has 7 or 8 developers without testers. They have 1 testing team, only testers in this team. Development team and testing team manage the team individually.
- Interviewee 2: I am a tester; I just check code and run applications to find bugs. More than 70 developers in the company and more than 12 people make up the test team, about 10 testers and 2 test team managers. There are several expertise teams. When developers finish their work, deliver to testers for testing. They work independently of each other.
- Interviewee 3: My main task is test execution. There are 15 developers in the organization. 6 people make up the test team, 6 testers, no team manager. The testers were managed by project team managers. They have several expertise teams. Three develop teams and one test team. The teams work independently. The test team completely follow the test team manager's leadership.
- Interviewee 4: I am a tester and I am responsible for test development and execution. There are 20 developers in the organization. In a test team, there are 4 testers and no team manager. We have 24 people in the team, one manager, 20 developers, and 3 testers. The manager manages developers and testers together. The test team completely follow the test team manager's leadership.
- Interviewee 5: It is a small-sized company. I am a test team manager and responsible for test management. They put 20 persons on market. 15 product managers, 9 UI designers, 40 developers (20 iOS, 20 Android), 9 persons in test team (1 manager and 8 testers). The company has functional system and project system. In functional system, they divided the employees by their responsibility of the project. Test manager is responsible for the allocation of testers. In project system, each project team include project manager, developer, tester. Department manager and project manager manage the team together. Test manager is responsible for the allocation of testers. Project manager is responsible for managing the project team. test manager and project manager can be able to manage the tester together. To improve the level of test team, the inexperienced testers often study with experienced testers.
- Interviewee 6: They put 20 persons on market. 15 product managers, 9 UI designers, 40 developers, 9 persons in test team (1 manager and 8 testers). The company has two systems: functional system and project system. In functional system, they divided the employees by their responsibility of the project. In project system, each project team include project manager, developer, tester and so on.
- Interviewee 7: There are 15 developers and 3 testers (1 team manager and 2 testers) in this organization. In a project team, the developers and testers work together, the project manager has responsible for all the people in the team. There is different working way in different condition. The developers and testers could work together or dependently.
- Interviewee 8: 15 developers in the organization. There are 3 people in test team, 1 team manager and 2 testers. The testers work together with developers. The company has separated develop team and test team. Depending on the duration of different project, they take appropriate solution and amount of task. This is not a fixed process. The team manager assigns the tasks based on the level and workload.
- Interviewee 9: There are 40 developers in this organization. And 0 test manager and 2 tester in test team. In a project, the testers and developers work together from the first requirements analysis phase to the re-test phase. The test team and developers follow the project manager's leadership. The test team manager is responsible for allocating the task and submitting daily report.
- Interviewee 10: The development team has about 30 people. There are 6 people in test team, 1 manager and 5 testers. One of the tester is responsible for hardware testing. The test team manager allocates tasks for testers. The develop team and test team work together in a project. The test team manager has absolute power for the test team, the testers must follow the team manager's leadership.
- Interviewee 11: I have been involved in test management activities and I work for testing execution. The company has 30 developers in this organization, 6 people of test team, 1 team manager and 5 testers. A project team includes developers and testers, they work together.
- Interviewee 12: There are 40 people in develop team, 38 developers and 2 testers (1 tester and 1 team manager). The 38 developers have 1 manager which has responsible for the attendance and task allocation. The 38 developers have been divided into 3 project team for different project. The project teams work independently. After the development, they would give the project to tester, and the test team manager only has responsible for tester. The developers and testers work independently.
- Interviewee 13: There are 40 people in the develop team. 38 developers and 2 testers. 1 manager in 38 developers and 1 test team manager in test team. The 38 developers have been divided into 3 project team. And test team has responsible for all the project testing.

Figure 4.3: The interview result for RQ1

Table 4.5: The percentage of developers in software startups and the testers in development team

Company name	Company size	Development team size	The percentage of development team in company	Test team size	The percentage of test team in development team
ShanDong Inspur software	160	50	31.3%	6	12%
Kingsoft software	150	70	46.7%	12	17.1%
ShanDong Cvices software	100	15	15%	6	40%
Hisense Group software	42	20	47.6%	4	20%
True love network	110	40	36.4%	9	22.5%
Beijing qingfeng software	65	15	23.1%	3	20%
Beijing Interactive streaming	150	40	26.7%	2	5%
Panda electronics	120	30	25%	6	20%
Hangzhou Upyun software	80	40	50%	2	5%

Table 4.6: The percentage of test team manager in test team

Company name	Test team size	The number of test team manager	The percentage of test team manager in test team
ShanDong Inspur software	6	1	16.7%
Kingsoft software	12	2	16.7%
ShanDong Cvices software	6	0	0
Hisense Group software	4	0	0
True love network	9	1	11.1%
Beijing qingfeng software	3	1	33.3%
Beijing Interactive streaming	2	0	0
Panda electronics	6	1	16.7%
Hangzhou Upyun software	2	1	50%

Based on the data from table 4.5, we found that the percentage of developers in the 9 companies we interviewed concentrated about 30%. And the percentage of testers in develop team is about 20% in the 9 software startups.



Based on the data from table 4.6, in the 9 software startups, most of companies only have 1 test team manager, one company has 2 test team managers. And three companies do not have test team manager, because the three companies do not have separate test team, the testers were assigned into different project team, and they were managed by project team manager.

We statistics the way of test team management in the 9 software startups, we found there are three styles to manage test team in the 9 companies. And it shows in Table 4.7.

Table 4.7: The management style of 9 software startups

Name of software startups	Management style
ShanDong Inspur software	Style 2
Kingsoft software	Style 2
ShanDong Cvices software	Style 1
Hisense Group software	Style 1
True love network	Style 3
Beijing qingfeng software	Style 3
Beijing Interactive streaming	Style 1
Panda electronics	Style 2
Hangzhou Upyun software	Style 2

Note:

Style 1: The company includes several project teams. Each team includes several developers and testers. They work together. Testers are managed by project manager.

Style 2: The company includes several functional departments. It has separate test department. Developers and testers work independently. Testers are managed by test team manager.

Style 3: The company has both project team and test department. Testers are managed by project manager and test team manager.

Based on the analysis of the result from Table 4.7, we found 3 companies are style 1, accounting for 33.3%, 4 companies are style 2, accounting for 44.4%, 2 companies are style 3, accounting for 22.2%. The proportion of the 3 styles in the 9 companies is close. We also found that style 3 are a combination of style 1 and style 2. So, we think testers in software startups often managed by test team manager, or assigned into different project team, and managed by project manager.

## 4.2 RQ2: What test processes and test techniques software startups follow to ensure quality of end products?

### 4.2.1 Result and analysis of literature review

13 of the 20 literatures are related to the test process and techniques of regular software companies. Table 4.8 shows some information about the 13 literatures.

Table 4.8: Basic information of 13 literatures related to RQ2

Title of literature	Author(s)	Published year	Research area	Source	Reference
A comparative study of	Auju Bansal	2014	Software	Journal	[67]

software testing techniques.			testing		
A walk through of software testing techniques	JaiGaur, Akshita Goyal	2016	Software testing	Conference article	[1]
Software testing techniques: A literature review	Muhammad Abid Jamil	2016	Software testing	Conference article	[68]
Software testing techniques	ShivKuamr Hasmukhrai Trivedi	2012	Software testing	Journal	[69]
A comparative study of manual and automated testing for industrial control software	Eduard Enoiu	2017	Software testing	Conference article	[70]
A software function testing method based on data flow graph	Wenjing Cao	2008	Software testing	Conference article	[71]
Modeling software testing process using HTCPN	Lizhi Cai, Weiqin Tong	2009	Software testing	Conference article	[72]
A Embedded software testing process model	Huaming Qian	2009	Software testing	Conference article	[73]
Evaluation and measurement of software testing process quality applicable to software testing laboratory	Yang Sun	2015	Software testing	Conference article	[74]
Identifying a subset of TMMI practices to establish a streamlined software testing process	Kamilla Gomes Camargo	2013	Software testing	Conference article	[75]
Agile approach for software testing process	Rijwan Khan	2016	Software testing	Conference article	[76]
Analysis of black box software testing techniques: A case study	Mumtaz Ahmad Khan	2011	Software testing	Conference article	[77]
Graybox software testing methodology embedded software testing technique	Andre'c coulter	1999	Software testing	Conference article	[78]

8 of the 13 literatures are related to the test techniques in software companies. Literature [67] introduced white-box testing, black-box testing and grey-box testing. It showed the characteristics, advantages and disadvantages of each test techniques. It also provided the which situation the three techniques are suitable for. Literature [1] described four test strategies: unit testing, integrating testing, system testing and acceptance testing. It also introduced the test techniques, and to help know more about these test techniques, the literature illustrated with examples for the techniques. In literature [68], it simply introduced the three test techniques and the

lifecycle of testing. Literature [82] introduced many different types of tests and the suitable testing techniques. For example, the function testing is suitable for using black-box testing. Then the literature mainly described black-box and white-box. The literature [70] analyzed the advantages and disadvantages of manual testing and automated testing, and tested the test coverage of both techniques through a case study. To help with function testing, literature [71] proposed a data flow graph method. It also indicated that, the function testing would ignore the internal structure of the program, because the tested program in function testing treats as a black-box. In the literature [77], two kinds of test methods: black-box testing and white-box testing were analyzed, and it presented a tool that can generate different test cases automatically for black-box testing. In literature [78], the author described that the grey-box testing is a test method for function testing, it combines the characteristics of black-box testing and white-box testing. In addition, grey-box testing can be used to automate code generation and automated software validation, to reduce the time, cost and effort of testing.

Based on these literatures, we found the commonly used testing techniques in most regular software companies are white-box testing, black-box testing and grey-box testing. To ensure the effectiveness of software testing, the companies often select the most suitable test techniques based on specific project. None of the software test techniques can be applied to all the software projects. We also found both manual testing and automated testing have advantages and disadvantages, and they cannot replace each other. The regular software companies often combine manual testing with automated testing.

Based on these findings, and there's few literatures related to the test techniques used in software startups, so we decided to investigate the test techniques of software startups.

6 of the 13 literatures are related to the test process in software companies. literature [68] not only discussed the test techniques but also introduced the test process. Table 4.9 shows the test process we summarized from 6 literatures.

Table 4.9: Summary of test process from literature review

Software test process	Reference
Requirement analysis→test planning→test case development→test case execution→test result reporting→defect retesting→regression testing→test closure	[68]
GO (test goal set)→RS (resource set)→AS (elementary active set)→PR (set of product such as test case)→RU (rule set in process)→ST (structure of process)→RO (set of testing activities)	[72]
Test application→test plan→test design→test execution→regression testing→result evaluating	[72]
Test plan→test plan review→test case design→test execution→test summary→test report→regression test	[73]
Requirement analysis→planning→design and implementation→execution→test report→reviews of testing phase	[74]
Test requirement analysis→test requirement review→test case design→test case review→test execution→test report→regression testing	[74]
Test planning→test case design→set up test environment and data→execution and evaluation→monitoring and control	[75]
Identify requirement→acceptance criteria→strategy planning and design→executing→defect tracking and bug fixing→review and audit→acceptance and baseline	[76]

Based on these test processes from literature review, we had a deep analysis and understanding of each step of test process, then merged and de-duplicated the steps, we summarized test process often includes 15 steps. Table 4.10 shows the 15 steps of test process.

Table 4.10: The steps of test process

ID	The steps of test process	Brief introduction
1	Requirement analysis	Testers are involved in software requirement analysis, to make the test plan.
2	Requirement review	The testers review the requirements analysis document to ensure they can understand the user's requirements.
3	Test plan	Test team develops test plan, which includes test duration, test range, etc.
4	Test plan review	Review the test plan again to ensure the test plan is effective and executable.
5	Test case design	Design the effective and executable test cases.
6	Write test case	Use standard test case templates to write test cases.
7	Test case review	Review the test cases not only based on test plan, but also based on requirements document.
8	Set up test environment	Set up hardware environment and software environment for testing.
9	Select test method and techniques	Select the appropriate test method and techniques according to the type of software project.
10	Test execution	Execute software testing step by step based on test plan and test cases.
11	Test record	Record the results of test execution, include the number of bugs, test duration, etc.
12	Defect tracking and bug fixing	Track the defect and ensure the bugs are eliminated.
13	Test report	Generate test report based on the analysis results of test data
14	Regression test	Retest after the old codes were modified, and confirm the modification was not lead into new bugs.
15	Test closure	Test closure, review the test work, summarize the shortcomings, and try to find some ways to improve the test.

Based on these findings from literature review, test process can be defined as a sequence of software test activities. It can seem as a roadmap for software testing which could reflect different test activities in different time [72]. So, the rationality of test process is important to the success of software testing. If only rely on the professional testers, but lack of proper test process, the testing has high risk to fail. Therefore, we decided to investigate the test process in software startups.

#### 4.2.2 Result and analysis of survey

We organized the interview results from 13 interviewees in the 9 software startups. And we found out related information about the test techniques and test process of the startups for RQ2. We marked important information with red color in Figure 4.4. Then based on the result of interview, we statistics and classified the test techniques in the 9 software startups. Table 4.11 shows the test techniques used in the 9 software startups. Table 4.12 shows the classification of test techniques based on test design. Table 4.13 shows the classification of test techniques based on manual

or automated testing. Table 4.14 shows the classification of test techniques based on function or non-function testing.

- Interviewee 1: I am a tester and mainly responsible for executing testing assignments. The process of testing in the company is: Requirement analysis → Test plan → Test design → Sort out test cases → The division of the test staff → Execution test plan → Write test documents. In the development process, the developer will use unit test. They don't need tester help. Before each release, the completed project will be handed over to the test team. They will use manual testing and Monkey testing. When they complete all the release of test, the product will be shelves. No acceptance tests. The common test techniques my company used is unit testing, function testing, black-box testing and monkey testing. Monkey test is an automated test method that does not require a lot of human resource. Unit test can find many illogical bugs or problems in development process. Manual testing can find some problems that the automatic testing cannot find. It has good robustness. Monkey test and unit test is popular in China. It can save time and money in start-up companies. In addition, the number of employees is limited, so we need to select the simple techniques.
- Interviewee 2: I have been worked 1 year as a tester. My main task is just check code and run applications to find bugs. The company produces mobile app, web and PC server with JAVA or C. We often start testing when the development finished and end testing before the project delivered. The main testing process is: requirements analysis, test design, write test case, executing, and checking. Dynamic testing, function testing, automated testing and black-box testing are the main testing techniques, because it can simulate the real environment.
- Interviewee 3: As a new tester in my company, my task is test execution. The project I have been participated in is a medical related app and a web system. Since white-box testing is in code level, we can do the test according to program logic, and most tester of the company have professional programming knowledge, my company often use white-box testing, black-box testing, automated testing, function testing, non-function testing and acceptance testing. The testing process begin at mid-term of project and over after the project delivered. The testing process includes: requirements analysis, test plan, test design, executing, checking.
- Interviewee 4: I have been worked as a tester in the company 2 years. I take part in test development and test execution. The testing techniques we used is black-box testing, automated testing and function testing. For better user experience and reduce the budget, Black-box testing can do the test based on the user experience, it can be improved the efficiency. Also, the repetitive work can be done with machines. Testing processes happened from development finished to the project coming to delivered. The main parts of test practice process are: requirements analysis, test plan, test design, write test case, executing, checking.
- Interviewee 5: As a test team manager, I have 5 years working experience. I am responsible for test management in this company. The organization starts testing when most of development work has completed and end testing after the project delivered. There are many steps of testing process: Requirements review, make project schedule, write test case, interface testing, large model test, integration testing, system test, acceptance test, product release. Developers do the interface testing after development without testers. Since the start-up company lack of testers, acceptance test usually be ignored. Some test tools are common used in my company: Jira (tools to manage bugs), Monkey testing, Python+ (open source architecture, use for interface layer automation platform implementation), Git (use for managing code), SVN (use for managing requirements document), Loadrunner jmeter (use for performance testing). Based on the current situation of the company, these tools are suitable for the project and for better user experience and reduce the budget.
- Interviewee 6: I am a tester in the dating website company. My task in the test team is test development and execution. My test team often use automated testing and black-box testing. There are some tools we often used: Project (tool to make project schedule), Jenkins (use for integration testing), Monkey, etc. There are some test practice processes: Requirements review, make project schedule, write test case, interface testing, large model test, integration testing, system test, acceptance test, product release.
- Interviewee 7: I have been worked as a test team manager in the company 2 years. My task is for the test team management. My company mainly develop the education related app in Android and IOS. The testing process begin after the requirements analysis and end after the product delivered. There could have online test after product delivered. Normally, testing could be end after the product delivered one week without serious bugs. The process of testing including test plan, test design, test case, test execute, and test report. Test report is occurred before the product delivered. The main testing techniques is automated testing, function testing and black-box testing.
- Interviewee 8: As a tester in company, we often have responsible for test execution, our team use JAVA to develop app in Android and IOS. The testing process begin after the requirements analysis and end after the product delivered. We use automated testing, function testing and black-box testing. We often use some tools, such as Monkey, Python+, etc.
- Interviewee 9: I am a test team manager. I have 10 years of software testing experience. I have been working in this company for 2 years. My main work is to assign different testers to different project groups. And I will assign different test tasks to different testers. The testing process is: Requirements analysis → Assign test tasks → Write test cases → Test case review → Test execution → Cross testing → Re-test. Our company generally uses black-box test, automated testing and function testing. While white-box testing and performance testing are important, we do not use white-box testing and performance testing due to money, experience and test knowledge level limitations. The test tools: We generally use free testing tools. For team management, we usually use a system management tool named "chandao". And for bug management, we always use some online apps to manage them. To improve team efficiency, I asked each tester to submit a report every day to see their daily work. And, we regularly train our testers to improve their testing level.

- Interviewee 10: I am a test team manager. I have 5 years of software testing experience. My job is software testing team management. And I have been working in this company for 2 years. The software testing process is: Requirements analysis->test plan->Write test cases according to the requirements analysis result->Test case review->Test execution->Regression Testing->Write test documentation. To facilitate communication with the developer, the company uses an online system called "chando". Testers and developers can interact online to facilitate bug feedback. I often assign the complex test tasks to experienced tester, and assigns the simple test tasks to inexperienced testers. Besides, to improve the level of test team, the inexperienced testers often study with experienced testers. We mainly use black-box testing, automated testing, function testing, non-function testing and manual testing. Because our company is a start-up company, the company version iterations are particularly fast, so automated testing is difficult to apply to start-up companies. The cost of automation is high. The test tools that our company often use are: Fildler (use for checking the returned data after testing), Project (tool to make project schedule), Monkey.
- Interviewee 11: I am a tester; I've been working in this company for two years. Although I have ever participated in the test management work, my main work is executing testing assignment. In our company, the testing work is generally started after the end of the development phase. And the testing work ends before the product is delivered. The process of testing in this company is: Requirement analysis-> Test plan-> Test design-> Execution test plan-> Test Verification->Re-test. Our company generally only uses black box testing. We do not apply white box testing. Because, we are start-up companies, the testers in our company lack of experience and knowledge. They cannot use white box testing. And to be honest, black box test can guarantee that our products will satisfied our customers' requirements, it is enough for start-up companies.
- Interviewee 12: I am a test team manager. I have 3 years of software testing experience. My job is software testing team management. I have been working in this company for 1 year. I am responsible for the distribution and regulation of testers. I allocate different number of testers into different project team. These testers are managed by their respective project team managers. I will allocate testers based on the level of the testers and amount of test work. I will pay attention to the skills of training testers, When the time is sufficient. The testing process: Requirements analysis->Test plan->Write test cases->Test case review->Test design->Test execution->Re-test. After the product is delivered, the test team can take a new test project. Our company mainly uses black-box testing, function testing, automated testing, also has some performance test and manual test. We do not use automated testing. Because the automated testing requires testers has a high level of experience and knowledge. And the cost of automated testing is great. However, for a start-up company, in early days of this company, usually do not have any very professional testers. The test tools that our company often use are: Road runner and Jmeter.
- Interviewee 13: I'm a tester. I have 1 year's software testing experience. I've been working in the company 1 year. My main job is to write test cases and execute them. In other words, it's test execution, rather than test management. My main test project is IOS products. The testing process is: Write test cases->Test case review->Test execution->Re-test. In test process, we divided bugs into (higher middle and lower) 3 levels. If there're 3 middle or higher-level bugs in the project, we don't recommend product to be delivered. I think write high-level test case is important for software test. We mainly uses black-box test, also has performance test and manual test. The test tools our company often use are: Road runner, Jmeter and some other online testing tools.

Figure 4.4: The interview result for RQ2

Table 4.11: Test techniques used in the 9 software startups

Name of software startups	Test techniques
ShanDong Inspur software	Black-box testing, automated testing, function testing
Kingsoft software	Black-box testing, dynamic testing, automated testing, function testing
ShanDong Cvices software	Black-box testing, function testing, white-box testing, automated testing, non-function testing
Hisense Group software	Black-box testing, function testing, automated testing
True love network	Black-box testing, function testing, automated testing
Beijing qingfeng software	Black-box testing, function testing, automated testing
Beijing Interactive streaming	Black-box testing, function testing, automated testing
Panda electronics	Black-box testing, function testing, automated testing, manual testing, non-function testing
Hangzhou Upyun software	Black-box testing, function testing, automated testing

Table 4.12: Classification of test techniques based on test design

Name of software startups	Test techniques		
	Black-box testing	White-box testing	Grey-box testing
ShanDong Inspur software	√		
Kingsoft software	√		
ShanDong Cvices software	√	√	
Hisense Group software	√		
True love network	√		
Beijing qingfeng software	√		
Beijing Interactive streaming	√		
Panda electronics	√		
Hangzhou Upyun software	√		

Table 4.13: Classification of test techniques based on manual or automated testing

Name of software startups	Test techniques	
	Manual testing	Automated testing
ShanDong Inspur software		√
Kingsoft software		√
ShanDong Cvices software		√
Hisense Group software		√
True love network		√
Beijing qingfeng software		√
Beijing Interactive streaming		√
Panda electronics	√	√
Hangzhou Upyun software	√	√

Table 4.14: Classification of test techniques based on function or non-function testing

Name of software startups	Test techniques	
	Function testing	Non-function testing
ShanDong Inspur software	√	
Kingsoft software	√	
ShanDong Cvices software	√	√
Hisense Group software	√	
True love network	√	
Beijing qingfeng software	√	
Beijing Interactive streaming	√	
Panda electronics	√	√
Hangzhou Upyun software	√	

Based on the data in the table 4.12, 4.13, 4.14, we found all the 9 software startups used black-box testing, only one company used white-box testing. We used interview to collect the data and found the black-box testing is the most commonly used method in the 9 software startups, because testers don't need to know the implementation details of the program code by using black-box testing. There are not many professional testers in software startups. The black-box testing method is simply, and don't need very high-level testers. It could save much time. Therefore, software startups use black-box testing typically. We also found these 9 companies are all used automated testing, only two companies used manual testing. Because automated testing can automate tedious test tasks and save much time for software

testing. Since most software startups don't want to spend too much time and money on testing, they often select automated testing. In addition, we found the 9 software startups all used function testing, only two companies used non-function testing. Because the limitation of test duration, most startups select function testing to test the basic function of program. They do security, usability and other non-function testing rarely.

Based on the interview results, we also found some tools used in software startups, most of these tools are automated testing tools, it has a great help to the testing of software startups. Table 4.15 shows these common testing tools

Table 4.15: Common testing tools used in 9 software startups

Number of tools	Name of tools	Brief introduction
No.1	Fillder	Use for checking the returned data after testing
No.2	Jira	Use for managing bugs
No.3	Monkey	Stress testing tool
No.4	Python+	Use for interface layer automation platform implementation
No.5	Jenkins	Use for integration testing
No.6	Git	Use for managing code and bug
No.7	SVN	Use for managing requirement and generating test plan
No.8	Loadrunner jmeter	Use for performance testing
No.9	“chandao”	A system testing tool
No.10	Road runner	Use for performance testing and stress testing of system
No.11	Jmeter	Use for stress testing

We investigated the test process of the 9 software startups, and result shows in Table 4.16. Then we analyzed each step of the test process, and statistics the number and proportion of the process in the 9 software startups. It shows in Table 4.17.

Table 4.16: Test process in 9 software startups

Name of software startups	The number of steps of test process	Test process
ShanDong Inspur software	7	Requirement analysis, Test plan, Test design, Sort out test cases, The division of the test staff, Execution test plan, Write test documents
Kingsoft software	5	Requirements analysis, Test design, Write test case, Executing, Checking
ShanDong Cvices software	5	Requirements analysis, Test plan, Test design, Executing, Checking
Hisense Group software	6	Requirements analysis, Test plan, Test design, Write test case, Executing, Checking
True love network	9	Requirements review, Make project schedule, Write test case, Interface testing, Large model test, Integration testing, System test, Acceptance test, Product release
Beijing qingfeng software	5	Test plan, Test design, Test case, Test execute, Test report



Beijing Interactive streaming	8	Requirements analysis, Test plan, Assign test tasks, Write test cases, Test case review, Test execution, Cross testing, Re-test
Panda electronics	9	Requirements analysis, Write test cases, Test plan, Test design, Test case review, Test execution, Regression Testing, Re-test, Write test documentation
Hangzhou Upyun software	7	Requirements analysis, Test plan, Write test case, Test case review, Test design, Test execution, Re-test

Table 4.17: Number and proportion of the process in the 9 software startups

The step of test process	The number of each steps in the software startups	The percentage of each steps
Requirement analysis	8	88.9%
Test plan/test design	8	88.9%
Write test case	7	77.8%
The division of the test staff	2	22.2%
Execution test	9	100%
Write test documents	3	33.3%
Checking	3	33.3%
Test case review	3	33.3%
Re-test	3	33.3%
Regression Testing	1	11.1%
Cross testing	1	11.1%

Through analyzing the results from the interviews, we found that there are four steps of software test process have a higher proportion. These four steps are arranged in descending order proportion, they are execution test, requirement analysis, test plan/test design, write test case. Besides, we found because the limitation of time, software startups do less requirement review, test case review and regression test.

### 4.3 RQ3: What are the testing challenges that software startups encounter in initial years?

#### 4.3.1 Result and analysis of literature review

3 of the 20 literatures are related to the testing challenges in regular software companies. Table 4.18 shows some information about the 3 literatures.

Table 4.18: Basic information of 3 literatures related to RQ3

Title of literature	Author(s)	Published year	Research area	Source	Reference
Organizational and customer related challenges of software testing: An empirical study in 11 software companies	Frank Philip seth	2014	Software testing	Conference article	[37]
What is software testing? And why is it so hard?	J.A.whittaker	2000	Software testing	Conference article	[83]
Software testing	Antonia	2007	Software	Conference	[2]

research: Achievements, challenges, dreams	Bertolino		testing	article	
---	-----------	--	---------	---------	--

Based on the literatures we found, we summarized the testing challenges come from three different stakeholders: tester, manager and customer.

#### **Challenges from tester:**

- Testers few involved in planning phase of software development, which could lead to underestimate the scope and resources of testing;
- Testers don't know much about the business area. To execute a high-quality testing, testers should have good development skills and know about the formal languages;
- Testers should specify the input of each interface, which include human interfaces, software interface, file system interface, communication interface. And they need to consider various available file formats, communication protocol;
- If the interface has problems about infinite size and complexity, testers must select the value of any variable input, not only the values of a single variable but also the combination of multiple variables;
- It is hard for testers to describe the bugs and defect accurately. If the descriptions are inaccurate, these bugs and defects are difficult to fix completely.

#### **Challenges from manager:**

- Project manager poor support for testing, that would lead to the testers few involved in some activities of software development;
- Project manager' behavior might lead to skip some important tests;
- Manager pay less attention to the test process.
- Determining when to stop testing is a challenge for manager, because the production of bugs and defects do not end with testing.

#### **Challenges from customer:**

- The willingness of customer to cooperate in after-sales testing could affect the efficiency of testing, and affect the quality of software;
- The limitation of the contracts of customer and software company;
- Budget constraints.

Based on the literature review, we also found some testing challenges except the three aspects above. Firstly, the structures of the organization and modes of operation could limit the quality assurance of software testing. And poor used available system cannot lead a reasonable resource management, which could lead to the conflict of team communication and requirements interpretation [37]. Then, the different execution scenarios of testing may impact the test results. Software testing require multiple scenario, but software companies don't set multiple test scenarios generally. It's a challenge for testing [83]. Besides, to have an effective test management, there must be a seamless interface to the latest system changes and business requirements [2].

Based on the findings of testing challenges in software companies, and there's few literatures related to the testing challenges in software startups, so we decided to investigate the testing challenges of software startups.

### **4.3.2 Result and analysis of survey**

We organized the interview results from 13 interviewees in the 9 software startups. And we found related information about the testing challenges of the startups for RQ3. We marked important information with red color in Figure 4.5.

Then based on the result of interview, we statistics and analyzed testing challenges in the 9 software startups. Besides, we investigated the causes of these testing challenges.

- Interviewee 1: As a tester in company, I think the main challenge in the testing process is that the product of development team has big difference with the requirements of customers and the communication between testing team and development team is difficult because the short of technologies they have. For the company, there's high economic pressure, so they give less investment for testing team. That lead to low activity of testing team. The construction of enterprise culture is not sound. The staff is unstable.  
To avoid these challenges: I think the company need to give more training of the test team and each team members. And the develop team should pay more attention to the test team to make sure the efficiency of the product and reduce more bugs.
- Interviewee 2: Most of testers lack of related knowledge and experience for the testing process. They are in different testing level and the team lack of specialized training, so the product often has some problems because of the low-quality testing process. And the test team manager doesn't have much time to manage the test team, so the efficiency of the test team is low. Another challenge of the company is funding problem. And the company lack of test backtracking, bug recurrence usually happened.  
To avoid the challenges: Training the test team and communication with developers. Good communication is necessary for the testers to know about product and improve testing process. Training makes the team at a same technical level, they can communicate easier.
- Interviewee 3: There are very few testers in my testing team, and test team lack of communication with develop team. The test team manager lack of leadership for the team members. For the startup company, they lack experience and they can't invest too much money. They could not put much time and effort for the test team, that lead to the product has many bugs.  
To improve the company and avoid the challenges: The team members should have more communications. The company need to select the manager with more experience, and allocate the testing tasks in detail. The test team manager should reduce the number of testers and improve the experience of testers.
- Interviewee 4: I am a tester, the main testing challenges in start-up companies are: Testers have low levels of expertise and cannot find the bugs in the project. The lack of communication between the various testers, repeated testing, useless test appeared frequently. Start-up company lack of money to recruit enough testers. Too many tasks but few testers, it is difficult to complete the task on time.  
How to improve it: To improve the probability of bug being discovered, the project is tested repeatedly. Focus on the communication within the test team to improve the efficiency of the test team. Improve the welfare of testers.
- Interviewee 5: I am a test manager, the main testing challenges are: Lack of enough professional employees. Start-up company don't have the complete test team in early year. Only 1 or 2 testers would do the testing tasks. Many testing tasks are completed by developer. Lack of enough development time. In general, the time for our company perform a test process is about 1 week. The reason is the manager is too anxious, just want to come up with products as soon as possible, they ignore the test process. Each stages of project development are incomplete. The developer could ignore some bugs in testing process. Since the funds of start-up company is limited, and the investor doesn't know more about the software industry, they think the testing in development is not important, so they put much money into other place. In early start-up company, team lacks cohesion. It's difficult to manage team. I need to know basic situation of each team members, their recent trends and moods. I should communicate with them frequently, solve the problems and eliminate the contradiction timely. In addition, the assessment reward should base on the current situation of each team members.  
How to improve it: To unite the team better, I should give some commitment to employees. If employees feel they work for their own career, they would pay more attention and responsibility to testing work. Also, the team should employ a capable man who is not only responsible for the work seriously, but also has positive influence and strong professional level to hold on team. And improve the training for tester to let them use testing tools expertly. It is easier to manage team members with high and professional level.
- Interviewee 6: I am a tester, the main testing challenges in start-up companies are: Lack of enough development time. In general, the time for our company perform a test process is about one week. The time that the project manager assigns to the software testers is always very few. Therefore, the testers will not conduct a system test for the product. The company is divided into functional team and project team, a person may be managed by two individuals at the same time causing some management conflict.  
How to improve it: If the start-up partner is a software development engineer or a software tester, he will invest a lot of effort to ensure quality. Because he (she) know that the quality of the software product will affect the users' satisfaction. And the level of user satisfaction directly affects the product sales.
- Interviewee 7: As a test team manager in startup company, our test team doesn't have clear functional division. The tester in my team always work together with me. The duration is short and the team lack of high level experienced human resource. Because of the human resource allocation of the company, I don't have specific leadership for the testers.  
To improve the testing process, as a team manager, I need to have organized leadership for test team. To ensure the testing process and improve the quality of the product, the company should pay more attention to the testing process and recruit some experienced tester.

- Interviewee 8: All the testers in different technique level, we always have different opinions with same task. And manager doesn't have specific activities for management. The company pay attention to development process and ignore the testing process. We have less tester for big project, and we don't have many useful tools or techniques for testing. It is hard to find bugs and repair bugs.  
I think company should allocate some experienced people training us. The manager should manage the team for more efficient work.
- Interviewee 9: I am a test manager, the main challenges are: Many leaders ignore the importance of testing. They believe that start-up companies don't need testers at early stage. Because start-up company are immature, they lack of sufficient experience to produce high quality software products. In other words, these companies are too anxious, just want to come up with products as soon as possible, ignoring the users' satisfaction and product quality. Software start-up companies lack experienced developers or technical experts, and sometimes testers find problems that developers cannot solve. In many software start-up companies, software testing process is beginning after product development is completed. Product development process and testing process is not normalized.  
How to improve it: Add some training, these training is mainly aimed at the leadership. The leader who participate in this training include: the manager of the development team, the manager of the test team, and some leaders of the project team. The main training content is about development process, the purpose of this training is to improve the leaders' understanding of software testing.
- Interviewee 10: I am a test manager, the main testing challenges are: Lack of sufficient testers. A project has a short test time. Testers are inexperienced and have low testing skills. In the software startup company, the customer's requirements change frequently.
- Interviewee 11: I am a tester, the main testing challenges in start-up companies are: In our company, there are only 3 testers, so lack of sufficient testers is the biggest challenge, testers are inexperienced and have low testing skills. In our company, the testers only have one year of test experience. Most of them only know what is black-box testing. Only one tester known how to conduct while-box test. These testers have little knowledge of the code.
- Interviewee 12: I am a test team manager, the main testing challenges in start-up companies are: Our company don't have the complete test team in early year. There only have few testers in our company. So, it is not a test team. It is just a few people check the programmer's work. Because it is not team work, so led to the lack of management. Where these testers lack of enthusiasm. Our company lack of money to recruit enough testers in initial years. Leaders do not attach importance to testing, they think the testing in development is not important, so they put much money into more important place. Our company lack of experienced developers or technical experts, and sometimes testers find problems that developers cannot solve it.
- Interviewee 13: I am a tester, the main testing challenges in start-up companies are: Most of testers lack of related knowledge and experience for the testing process. They are in different testing level and the team lack of specialized training, so the product often has some problems because of the low-quality testing process. The company often pay attention to development process and ignore the testing process. We have less tester for a big project, and we don't have many useful tools or techniques for testing.

Figure 4.5: The interview result for RQ3

Based on the interview from 9 software startups, we summarized the testing challenges into six points:

- The extensive test coverage of software products. In the 9 software startups, most of the companies only use simple black-box testing or function testing. Few companies have comprehensive testing for product. They lack non-function testing to ensure the quality of software product. So, the extensive test coverage is a challenge for the 9 software startups.
- A standardized and stable test process. The 9 software startups often change the test process based on different project or product. For example, sometimes test team participate into requirements analysis, but sometimes they write test cases directly according to requirement document. That could cause the important step to be omitted. So, a standardized and stable test process is a testing challenge for 9 software startups.
- Track and detect the bugs and defects, until they are completely fixed. The 9 companies rarely track and detect the bugs or defects. After the testers had completed the test, they would proceed to the next project test, they would not back to track or detect the bugs. It could lead to some bugs cannot be fixed before the product is delivered, and affect the quality of the product. So, track and detect the bugs or defects is a testing challenges for the 9 software startups.

- A standardized and reasonable management for test team. In the 9 software startups, sometimes all testers work together for a project, and sometimes they are assigned to different project teams. Testers in these companies can be managed by test team manager or project manager. The test team management is unstable. It could affect the effectiveness of testing. So, a standardized and reasonable management for test team is a testing challenge for the 9 software startups.
- Testers are familiar with testing methods and can use various testing tools skillfully. The 9 startups usually used black-box testing and some simple online tools. The testing level of testers is not high and it is limited by testing condition. Most of testers just can use black-box testing and some simple automated testing tools. That could make the testing difficult. So, testers are familiar with testing methods and can use various testing tools skillfully is a testing challenges for the 9 software startups.
- Testers and developers has unified opinion. Base on the investigation, we found that the 9 software startups pay attention to the delivery time of product. They want to publish the product in the shortest possible time. But fixing bugs could slow down the development progress, and lead to the delivery delayed. The developers are reluctant to fix bugs. And they cannot unify opinion with tester. It could affect development progress. So, tester and developers has unified opinion is a challenge for the 9 software startups.

From the interviews, we also found some reasons for the challenges. We list these reasons and give some descriptions in Table 4.19.

Table 4.19: Reasons of the testing challenges

Number of reason	Reasons	Brief description
No.1	Lack of investment in test team	Project manager gives less attention to testing, and they would not invest a lot in testing.
No.2	Lack of time	Software startups pay attention to delivery time of product, and leave little time for testing.
No.3	Lack of professional testers	Software startups generally underfunded and rarely hire professional testers.
No.4	Tester lack training	Software startups don't spend time and money on training for testers and developers. New employees are usually led by older employees
No.5	Lack of communication	Communication are not enough between developers and testers
No.6	Tester lack enthusiasm	Project manager do not pay much attention to software testing, which could lead testers lack enthusiasm, work inefficient
No.7	Test is constrained by development model	In different development model, test takes place differently. For example, test is after development in waterfall, test and development happened at the same time in agile
No.8	Developer replace tester	Because lack investment, tester is replaced by developer, it is difficult to find bugs
No.9	Frequent changed requirements	Frequent changed requirements lead to test case lose efficiency, and testing failed.
No.10	Misunderstanding of requirements	Testers are not often participated into requirement analysis, they just write test cases based on

		requirement documents, that lead to the misunderstanding of requirements
No.11	Bug recurrence	Lack of regression test, because fixing old bugs could produce new bugs.
No.12	Inappropriate test tools or techniques	Because the lack of professional testers, testers might select inappropriate test tools or techniques, which is not suitable for the project.

Based on these reasons from table 4.19, we found it could lead to the testing challenges. For example, No.2, 6, 8 could lead the low test coverage of software products. So, the extensive test coverage of software products become a testing challenge of software startups. Due to the characteristics of software startups, these challenges are normal. Therefore, we think the researchers should focus on the investigation of testing challenges, and propose some solutions to help software startups overcome these challenges, to improve the efficiency of testing.

## **5 DISCUSSION AND LIMITATION**

### **5.1 Discussion**

A comprehensive discussion of the results obtained from the two qualitative research methods, literature review and survey is provided in this section. We compared the interview results with the literature review results and try to validate them. From section 4, we use literature review and survey to conclude and analyze the results based on the research questions. We found out the company structure and management of the testing team, test processes and techniques, as well as the testing challenges in software companies by using literature review. And we got the information in software startups from survey. Based on these two qualitative research methods we selected for research questions, we firstly had a basic understanding of testing in software companies and some related knowledge about software startups. Then we knew more about the testing in software startups in practice. Hence, the research methods we selected is suitable for the study.

The research objects are come from China. Chinese software startups also have the common characteristics of software startups which are high growth and high risk co-exist. But we found that there are also some differences between Chinese software startups and other countries' software startups. The main difference is that software startups in China often have many people. Many Chinese software startups have more than one hundred people. It is rarely seen in software startups of other countries. Therefore, this research cannot represent all the software startups in the world, it can only provide current testing situation of a small part of Chinese software startups.

The research aims to explore the testing practice in software startups. We collected related data by using literature review and survey method. The data shows the testing practice in both software company and software startups. The reason we want to know about testing in software companies is that it could provide the basic current situation. Under the premise of testing practice in software companies, we can discuss the data and results from interviews by comparative analysis. The 9 companies we interviewed cannot represent all software startups, but we can know about test practices in software startups initially through the 13 interviews. Our study is a primary research to explore testing practices in 9 software startups, we provided some data and analysis results of the 9 companies to the researchers who want to research some related area. In addition, our research could help someone who plans to set up a software company. They can use the data we collected to think about the testing practice in their own company. Then find out the best way to prevent and resolve the problem in testing. Due to the combination and comparative analysis of the results, we discuss the findings related RQs from the two qualitative research methods.

For RQ1, the structure of a company involved in many aspects, we only focused on investigate the proportion of development team in software startups, and the proportion of test team in development team to show the company structure. Team management is complex in software company, so we investigated who the testers are managed by, to know about the team management. Based on the literature review, we found the percentage of developer and tester are different in different software companies. Regular software company often pay more attention to testing than software startups. Because we found in regular software companies, the number

of tester and developer is very close, and in some of these regular companies, the number of testers even more than developers. It can clearly be seen that regular company attach great importance to testing. According to the result of the interviews, we found in the 9 software startups, testers are far less than developers. Some of the startups do not have specialized testers, test was completed by developers. So, we think regular companies attach more importance to testing than software startups. Based on literature review about team management, we found seven ways of software company commonly used to manage the teams. Six of the ways is the traditional way of being led and managed by a leader. One of the ways is a new way of self-management. Compared to regular software companies, software startups have less testers, and test team management is relatively simple. From the interview, we found there are two ways for test team management in software startups. One way is don't set up test team manager, testers are assigned in different project teams, and managed by project team manager; the other way is to set up test team, testers are managed by test team manager.

For RQ2, it aims to identify the test techniques and test process used in software startups. In addition to the testing techniques or methods identified from the interviews, the testing techniques or methods identified from the literature review also be included, and both results would be compared. 28 testing techniques or methods were identified from the literature review, and they were compared to those identified from the interviews. The final list of testing techniques or methods after comparing and mapping the interview and literature review results is presented in Table 5.1. Based on the interviews, we found the 9 software startups took black-box testing as the most commonly used test techniques. They usually used automated testing to save much test time, and rarely used manual testing. Besides, we also found most software startups only used function testing, few startups used non-function testing to ensure the quality of testing.

In addition to software test process identified from the interview, the software test process from the literature review also be included, and both results would be compared. 15 main steps in the software testing process were identified from the literature review, and they were compared to those identified from the interviews. We merge the same steps and left 16 different steps. The main steps in the software testing process from literature review and interviews are assigned individual IDs for reference. The final list of the main steps in the software testing process after comparing and mapping the interview and literature review results is presented in Table 5.2. Based on interview, we found the test process in the 9 software startups usually include four steps: requirement analysis, test plan/design, write test case, execution test. Few software startups have requirement review, test case review and regression testing in the test process. Therefore, we think software startups don't have high demand for the quality of testing.

Table 5.1: Testing techniques or methods identified from literature review and interview

ID	Testing techniques/method	Source
No.1	Black box testing	LR, Interview
No.2	White box testing	LR, Interview
No.3	Gray box testing	LR
No.4	Manual test	LR, Interview
No.5	Automatic test.	LR, Interview
No.6	Unit test	LR
No.7	Functional test	LR
No.8	Integration test	LR



No.9	Scenario test	LR
No.10	System test	LR
No.11	Alpha test	LR
No.12	Beta test	LR
No.13	Stress test	LR
No.14	Load test	LR
No.15	Performance test	LR
No.16	Accessibility test	LR
No.17	Localization/globalization test	LR
No.18	Compatibility test	LR
No.19	Configuration test	LR
No.20	Usability test	LR
No.21	Security test	LR
No.22	Smoke test	LR
No.23	Build verification test	LR
No.24	Acceptance test	LR
No.25	Regression test	LR
No.26	Ad hoc test	LR
No.27	Sanity test	LR
No.28	Dynamic test	Interview

Table 5.2: The main test steps from literature review and interview

ID	The main steps in the software testing process	Source
No.1	Requirement analysis	LR, Interview
No.2	Requirement review	LR
No.3	Test plan/design	LR, Interview
No.4	Test plan review	LR, Interview
No.5	Write test case	LR, Interview
No.6	Test case review	LR, Interview
No.7	The division of the test staff	LR, Interview
No.8	Set up a test environment	LR
No.9	Select test techniques or methods	LR
No.10	Execution test	LR, Interview
No.11	Test record	LR, Interview
No.12	Defect tracking and bug fixing	LR
No.13	Test report	LR
No.14	Regression Testing	LR, Interview
No.15	Test closure	LR
No.16	Cross testing	Interview

For RQ3, we summarized and analyzed some testing challenges in software companies and software startups based on the research methods. We found similarities and differences of some existing testing challenges in software companies and software startups. Most testing challenges in regular company are come from different stakeholders, such as testers, managers, customers. Most of these challenges are man-made. The testing challenges of software startups are mostly because of the characteristics of software startups. For example, the willingness of customer to cooperate in after-sales testing could affect the efficiency of testing, and affect the quality of software, it is a testing challenges in regular software company; because lack of investment and time, lead to the low test coverage. It is a testing challenges of software startups.

## 5.2 Limitations and Validity threats

### 5.2.1 Limitations

Even though the research results had answered the research questions, this study still had some limitations. The limitations of the research are listed below:

- We only selected 9 software startups and 13 interviewees as the research object. The quantity is too few. And sampling method we selected is convenience sampling, it cannot represent the target sample. Due to the limitation of the number of research object, it is too few to represent the target population. We cannot come up with some quantitative conclusions. The research only helps us to simply learn about the test practices in a little part of software startups.
- The objects and interviewees are from China. There are some differences between Chinese software startups and other countries' software startups, so the research cannot represent software startups all the world. Besides, because of the limitation of the number of research object, 9 software startups also cannot represent all Chinese software startups. We would expand the research and investigate more companies all around the world.
- As we know, software testing can be divided into verification and validation. Verification aims to check or test the system by evaluating results of pre-specified requirements [69]. Validation aims to validate the correctness of the system, which is checking the specified content is what the user really wants or not. This research mainly focused on verification process, and lack of validation. To research testing practices in software startups, we should combine verification and validation. We'll pay more attention to validation in future research.
- The research proposed some testing challenges in software startups. Because the limitation of time and knowledge level of the two researchers, we did not propose any effective ways to solve these testing challenges in software startups. We would do more research to know about the testing challenges, as well as the solution of the testing challenges.

### 5.2.2 Validity Threats

In this part, we discuss some of the validity threats encountered in the research. Validity analysis is a process that identifies the imminent validity threats and tries to find some measures to mitigate these threats [84]. We identified some of the validity threats which affect the execution of the research. We analyzed the validity threats from four aspects: internal validity, external validity, construct validity, conclusion validity.

#### 5.2.2.1 Internal validity

There is an internal validity threat encountered in the research, it is how to collect and analyze the data in an appropriate or accurate way. Because both of us are not experts in the field of the research, this threat was mitigated by continuously validating the analysis with our supervisor. And this threat also can be mitigated by the following means. Firstly, we need to do as much as possible literature review to understand the area of the research. Then through the discussion with our supervisor to determine the optimal interview questions, do not ask some unnecessary questions. During the interview, we not only need to ask the interviewees some questions that have been identified before, but also need to ask some open questions according to the actual situation. When we collect and analyze the data, we also conducted follow-

up interviews with the interviewees to ensure the reliability of the interview results.

#### **5.2.2.2 External validity**

External validity of the research is the risk about the population selection of the interview. There is a risk involved in this area because the researchers need to look for the people who are working in software startups. And they need to be familiar with software testing process. So, looking for the interviewees could be a problem. To mitigate this problem, at the beginning of the research, the researchers used Google search engine to find a lot of Chinese software startups. Then sent e-mails to the test managers or testers of these companies, in order to get their consent to conduct our interview. After getting the consent, the researchers went to these companies, to know about some basic information of the interviewees before the interview, and it is also to know the interviewees could answer the interview questions or not. Then the researchers excluded some interviewees who are not helpful to the research. Through selecting eligible interviewees to mitigate these threats, it would help to get more useful information from the interviews.

#### **5.2.2.3 Construct validity**

There is a construct validity threat of the research that is the result of the research is not reliable. In other word, it is involved in specific case of the research. Because of lack adequate time, we could not contact as much as possible interviewees to participate in the interviews. And because of some other reasons, a part of interviewees was excluded. So, we only conduct a limited number of interviews, which includes 13 interviewees from 9 software startups. We could not mitigate the threats, and if we had a chance to perform more interviews, we believe that the research result will become more reliable.

#### **5.2.2.4 Conclusion validity**

There is a conclusion validity threat that was the possibility to get wrong conclusions from the result of research. Respondents who are involved in the interviews may form the conclusion validity threat. Because the respondents are in different educational level, and their work experience is not same, the degree of understanding of the interview questions could not be same. To mitigate this threat and help the interviewees have a clear understanding of the research questions, in each interview, we asked the interviewees with open questions, we conducted a practical statistical analysis of each interview from each interviewee. For the research questions, we used effective qualitative and quantitative data analysis methods to find the appropriate answer.

## 6 CONCLUSION AND FUTURE WORK

### 6.1 Conclusion

With the rapid development in the software industry, innovative software startups became an increasing trend in software market. Software startups can use a few resources to quickly develop and market innovative software products [5]. For the success of software startups, the product of software startups should be accepted by users. As we know, a high-quality software product could improve the user's satisfaction, so software testing is important for software quality assurance [29]. And we found many different regular software companies spend up to 40-50% of development efforts on software testing [29] [2], but time and money are finite in software startups, most software startups failed because run out of cash [11]. We are not sure that software startups would spend 40-50% of development efforts on software testing. We think that the investment of regular software company and software startups is different. So, the testing practices may different in regular software companies and software startups. But few research studies have investigated the testing practices in software startups. Therefore, we decided to conduct an exploratory study to know about the testing practices in software startups. Based on our research aim and objectives, we formulated three research questions to perform the research. In this study, we used literature review and survey method to find out the answers of the research questions. Totally, we interviewed 13 participants from 9 Chinese software startups. Here are the findings:

For RQ1, it is about the structure of software startups and how they manage the test team. We found that (1) the percentage of developers in the 9 companies we interviewed concentrated about 30%. (2) Although developers and testers in a develop team are equally important based on the literature review, the test process in most software startups still not be paid attention to. The percentage of testers in develop team is about 20% in the 9 software startups. (3) The 9 software startups often used two ways to manage the test team. One way is don't set up test team manager, testers are assigned in different project teams, and managed by project team manager; the other way is to set up test team, testers are managed by test team manager.

For RQ2, it is about the test processes and test techniques used in software startups. From interviews, we know about the test techniques used in the 9 software startups, why they selected these techniques and how they used these techniques. We also know about the test process in the 9 software startups. We found that (1) the 9 software startups often use the black-box testing, and automatically do some function testing. (2) Few of the 9 software startups used non-function testing to ensure the quality of testing. And because of the lack of professional testers, the software startups rarely used white-box testing. (3) The test process in the 9 software startups usually include requirement analysis, test plan/design, write test case and execution test. Few software startups have requirement review, test case review and regression testing in the test process.

For RQ3, it is about the testing challenges that software startups encounter in initial years. we found some different answers by interviewing the participants of the 9 software startups. Based on the interviews of 9 software startups, we summarized the testing challenges into six points: (1) The extensive test coverage of software products. (2) A standardized and stable test process. (3) Track and detect

the bugs and defects, until they are completely fixed. (4) A standardized and reasonable management for test team. (5) Testers are familiar with testing methods and can use various testing tools skillfully. (6) Testers and developers has unified opinion.

After the discussion in section 5.1, we found regular software company often pay more attention to testing than software startups. In initial years of business, software startups do not put too much money and time on testing generally. The test process in software startups is simple. The test techniques used in software startups is only a few simple functional test techniques. Therefore, we think software quality is not the most concern for software startups. Quickly put the product into the market and gain the benefits is their main purpose.

This study is a primary research to explore testing practices in 9 software startups, the contribution of the study is that our research could provide some data and analysis results of the 9 software startups to the researchers who want to research some related area. In addition, our research could help someone who plans to set up a software company. They can use the data we collected to think about the testing practice in their own company, and know about some testing challenges in advance. Then find out some ways to prevent and resolve the problem in testing. The research opens the door to research area about the test practices in software startups. We hope we can inspire more people to participate in the research of testing practices in software startups. Without spending too much time and money, improve the success rate of software startups by optimizing the testing.

## **6.2 Future work**

The future work of the research includes the following points:

- The research proposed some testing challenges in software startups. But did not propose any effective ways to solve these testing challenges. We would like to propose a systematic, detailed improvement strategy that can fundamentally reduce or eliminate the impact of these challenges.
- This research has only conducted a research in 9 Chinese software startups, in the future, we will investigate more software startups, not only in China, also in other countries, to find more data to support the research conclusions.
- As we know, software testing can be divided into verification and validation process. This research mainly focused on verification process, and lack of validation. We'll do more research about validation in future, to support and improve the research conclusions.

## REFERENCES

- [1] Gaur, Jai, et al. "A walk through of software testing techniques." System Modeling & Advancement in Research Trends (SMART), International Conference. IEEE, 2016.
- [2] Bertolino, Antonia. "Software testing research: Achievements, challenges, dreams." 2007 Future of Software Engineering. IEEE Computer Society, 2007.
- [3] Giardino, Carmine, et al. "What do we know about software development in startups?" IEEE software 31.5 (2014): 28-32.
- [4] Giardino, Carmine, et al. "Key challenges in early-stage software startups." International Conference on Agile Software Development. Springer, Cham, 2015.
- [5] Unterkalmsteiner, Michael, et al. "Software Startups—A Research Agenda." e-Informatica Software Engineering Journal 10.1 (2016).
- [6] Sutton, Stanley M. "The role of process in software start-up." IEEE software 17.4 (2000): 33-39.
- [7] Blank, Steve. "Embrace failure to start up success." Nature News 477.7363 (2011): 133-133.
- [8] Blank, Steve. The four steps to the epiphany: successful strategies for products that win. BookBaby, 2013.
- [9] Chorev, Schaul, and Alistair R. Anderson. "Success in Israeli high-tech start-ups; Critical factors and process." Technovation 26.2 (2006): 162-174.
- [10] Luo, Lu. "Software testing techniques." Institute for software research international Carnegie mellon university Pittsburgh, PA 15232. 1-19 (2001): 19.
- [11] Insights, C. B. "The top 20 reasons startups fail." October 7 (2014): 2014.
- [12] Kaner, Cem. "Exploratory Testing, Florida Institute of Technology." Quality Assurance Institute Worldwide Annual Software Testing Conference, Orlando, FL. 2006.
- [13] Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley & Sons, 2011.
- [14] PATTON, Ron. "Software Testing. Edit." (2001).
- [15] Ernst, Michael D. "Static and dynamic analysis: Synergy and duality." WODA 2003: ICSE Workshop on Dynamic Analysis. 2003.
- [16] Ramler, Rudolf, and Klaus Wolfmaier. "Economic perspectives in test automation: balancing automated and manual testing with opportunity cost." Proceedings of the 2006 international workshop on Automation of software test. ACM, 2006.
- [17] Afzal, Wasif, Richard Torkar, and Robert Feldt. "A Systematic Mapping Study on NonFunctional Search-based Software Testing." SEKE. Vol. 8. 2008.-
- [18] Runeson, Per. "A survey of unit testing practices." IEEE software 23.4 (2006): 22-29.
- [19] Jorgensen, Paul C., and Carl Erickson. "Object-oriented integration testing." Communications of the ACM 37.9 (1994): 30-38.
- [20] Beizer, Boris. Software system testing and quality assurance. Van Nostrand Reinhold Co., 1984.
- [21] Schach, Stephen R. Object-oriented and classical software engineering. Boston: McGraw-Hill Higher Education, 2011.
- [22] Gage, Deborah. "The venture capital secret: 3 out of 4 start-ups fail." Wall Street Journal 20 (2012).
- [23] Crowne, Mark. "Why software product startups fail and what to do about it. Evolution of software product development in startup companies." Engineering Management Conference, 2002. I EMC'02. 2002 IEEE International. Vol. 1. IEEE, 2002.
- [24] Nguyen-Duc, Anh, Pertti Seppänen, and Pekka Abrahamsson. "Hunter-gatherer cycle: a conceptual model of the evolution of software startups." Proceedings of the 2015 International Conference on Software and System Process. ACM, 2015.
- [25] Paternoster, Nicolò, et al. "Software development in startup companies: A systematic mapping study." Information and Software Technology 56.10 (2014): 1200-1218.
- [26] Glass, Robert L. "The Standish report: does it really describe a software crisis?" Communications of the ACM 49.8 (2006): 15-16.
- [27] Basili, Victor R., and Barry T. Perricone. "Software errors and complexity: an empirical investigation." Communications of the ACM 27.1 (1984): 42-52.

- [28] Krishnan, Mayuram S., et al. "An empirical analysis of productivity and quality in software products." *Management science* 46.6 (2000): 745-759.
- [29] Ahamed, S. S. "Studying the feasibility and importance of software testing: An Analysis." *arXiv preprint arXiv:1001.4193*(2010).
- [30] de Vries, René G. "Towards formal test purposes." (2001).
- [31] Gelperin, David, and Bill Hetzel. "The growth of software testing." *Communications of the ACM* 31.6 (1988): 687-695.
- [32] Hayes, Nicky. *Successful team management*. Thomson Learning Emea, 1997.
- [33] Scott-Young, Christina, and Danny Samson. "Project success and project team management: Evidence from capital projects in the process industries." *Journal of Operations Management* 26.6 (2008): 749-766.
- [34] Burnstein, Ilene. *Practical software testing: a process-oriented approach*. Springer Science & Business Media, 2006.
- [35] Beizer, Boris. *Software testing techniques*. Dreamtech Press, 2003.
- [36] Crnkovic, Ivica. "Component- based software engineering—new challenges in software development." *Software Focus* 2.4 (2001): 127-133.
- review: a step-by-step approach." *British journal of nursing* 17.1 (2008): 38-43.
- [37] Seth, Frank Philip, Ossi Taipale, and Kari Smolander. "Organizational and customer related challenges of software testing: An empirical study in 11 software companies." *Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on*. IEEE, 2014.
- [38] G. Liebel, E. Alegroth and R. Feldt, "State-of-Practice in GUI-based System and Acceptance Testing: An Industrial Multiple-Case Study", 39th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), Santander, pp. 17 – 24, 2013.
- [39] Cronin, Patricia, Frances Ryan, and Michael Coughlan. "Undertaking a literature
- [40] Beyea, Suzanne C., and Leslie H. Nicoll. "Writing an integrative review." *AORN journal* 67.4 (1998): 877-880.
- [41] Fraenkel, Jack R., and Norman E. Wallen. *How to design and evaluate research in education*. McGraw-Hill Higher Education, 2003.
- [42] Keele, Staffs. "Guidelines for performing systematic literature reviews in software engineering." Technical report, Ver. 2.3 EBSE Technical Report. EBSE. sn, 2007.
- [43] Galvan, Jose L., and Melisa C. Galvan. *Writing literature reviews: A guide for students of the social and behavioral sciences*. Routledge, 2017.
- [44] Griffiee, Dale T. "Research tips: Interview data collection." *Journal of Developmental Education* 28.3 (2005): 36.
- [45] Babbie, Earl R. *Survey research methods*. Wadsworth, 1973.
- [46] Fowler Jr, Floyd J. *Survey research methods*. Sage publications, 2013.
- [47] Kumar, Sameer, and Promma Phrommathed. *Research methodology*. Springer US, 2005.
- [48] Kothari, Chakravanti Rajagopalachari. *Research methodology: Methods and techniques*. New Age International, 2004.
- [49] London, Kerry, and Michael Ostwald. "Architectural Research Methods." *Nexus Network Journal* 6.1 (2004): 51-53.
- [50] Brinkmann, Svend. "Interview." *Encyclopedia of Critical Psychology*. Springer New York, 2014. 1008-1010.
- [51] Brenner, Michael. *The research interview, uses and approaches*. Academic Press, 1985.
- [52] Flick, Uwe. *An introduction to qualitative research*. Sage, 2014.
- [53] Axelrod, Robert, ed. *Structure of decision: The cognitive maps of political elites*. Princeton university press, 2015.
- [54] Pulakos, Elaine D., and Neal Schmitt. "Experience - based and situational interview questions: Studies of validity." *Personnel Psychology* 48.2 (1995): 289-308.
- [55] Wrigley, Mike. "Tom Wengraf (2001). *Qualitative Research Interviewing: Biographic Narrative and Semi-Structured Method*." *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*. Vol. 3. No. 4. 2002.
- [56] Galletta, Anne. *Mastering the semi-structured interview and beyond: From research design to analysis and publication*. NYU Press, 2013.
- [57] Turner III, Daniel W. "Qualitative interview design: A practical guide for novice investigators." *The qualitative report* 15.3 (2010): 754.

- [58] Seaman, Carolyn B. "Qualitative methods in empirical studies of software engineering." *IEEE Transactions on software engineering* 25.4 (1999): 557-572.
- [59] Kvale, Steinar. *Doing interviews*. Sage, 2008.
- [60] Csikszentmihalyi, Mihaly, and Reed Larson. "Validity and reliability of the experience-sampling method." *Flow and the foundations of positive psychology*. Springer Netherlands, 2014. 35-54.
- [61] Tansey, Oisín. "Process tracing and elite interviewing: a case for non-probability sampling." *PS: Political Science & Politics* 40.4 (2007): 765-772.
- [62] Tongco, Ma Dolores C. "Purposive sampling as a tool for informant selection." *Ethnobotany Research and Applications* 5 (2007): 147-158.
- [63] Morse, Janice M., et al. "Verification strategies for establishing reliability and validity in qualitative research." *International journal of qualitative methods* 1.2 (2002): 13-22.
- [64] Brown, G. P. "Alan Bryman and Duncan Cramer, *Quantitative Data Analysis For Social Scientists*." *CANADIAN JOURNAL OF SOCIOLOGY* 21 (1996): 434-435.
- [65] Miles, Matthew B., and A. Michael Huberman. *Qualitative data analysis: An expanded sourcebook*. sage, 1994.
- [66] Itkonen, Juha. "Empirical studies on exploratory software testing." (2011).
- [67] Bansal, Anju. "A Comparative Study of Software Testing Techniques." *the proceedings of International Journal of Computer Science and Mobile Computing* 3.6 (2014): 579-584.
- [68] Jamil, Muhammad Abid, et al. "Software Testing Techniques: A Literature Review." *Information and Communication Technology for The Muslim World (ICT4M)*, 2016 6th International Conference on. IEEE, 2016.
- [69] Trivedi, Shivkumar Hasmukhrui. "Software testing techniques." *International Journal of Advanced Research in computer science and software Engineering* 2.10 (2012).
- [70] Enoiu, Eduard, et al. "A Comparative Study of Manual and Automated Testing for Industrial Control Software." *Software Testing, Verification and Validation (ICST)*, 2017 IEEE International Conference on. IEEE, 2017.
- [71] Wenjing, Cao, and Xu Shenghong. "A Software Function Testing Method Based on Data Flow Graph." *Information Science and Engineering*, 2008. ISISE'08. International Symposium on. Vol. 2. IEEE, 2008.
- [72] Cai, Lizhi, et al. "Modeling Software Testing Process Using HTCPN." *Frontier of Computer Science and Technology*, 2009. FCST'09. Fourth International Conference on. IEEE, 2009.
- [73] Qian, Hua-ming, and Chun Zheng. "A embedded software testing process model." *Computational Intelligence and Software Engineering*, 2009. CiSE 2009. International Conference on. IEEE, 2009.
- [74] Sun, Yang, Hongbing Qian, and Xiaochen Liu. "Evaluation and Measurement of Software Testing Process Quality Applicable to Software Testing Laboratory." *Software Engineering Conference (APSEC)*, 2015 Asia-Pacific. IEEE, 2015.
- [75] Camargo, Kamilla Gomes, Fabiano Cutigi Ferrari, and Sandra Camargo Pinto Ferraz Fabbri. "Identifying a subset of TMMi practices to establish a streamlined software testing process." *Software Engineering (SBES)*, 2013 27th Brazilian Symposium on. IEEE, 2013.
- [76] Khan, Rijwan, Akhilesh Kumar Srivastava, and Dilkeshwar Pandey. "Agile approach for Software Testing process." *System Modeling & Advancement in Research Trends (SMART)*, International Conference. IEEE, 2016.
- [77] Khan, Mumtaz Ahmad, and Mohd Sadiq. "Analysis of black box software testing techniques: A case study." *Current Trends in Information Technology (CTIT)*, 2011 International Conference and Workshop on. IEEE, 2011.
- [78] Coulter, Andre' C. "Graybox software testing methodology: embedded software testing technique." *Digital Avionics Systems Conference*, 1999. Proceedings. 18th. Vol. 2. IEEE, 1999.
- [79] Moe, Nils Brede, Torgeir Dingsøy, and Tore Dybå. "Overcoming barriers to self-management in software teams." *IEEE software* 26.6 (2009).
- [80] Jianshe, Li, and Li Ting. "Choice of organizational structure for software companies in growth stage." *2010 IEEE International Conference on Software Engineering and Service Sciences*. 2010.
- [81] Nagappan, Nachiappan, Brendan Murphy, and Victor Basili. "The influence of organizational structure on software quality." *Software Engineering*, 2008. ICSE'08. ACM/IEEE 30th International Conference on. IEEE, 2008.



- [82] Barrett, Stephen, et al. "Software development process." U.S. Patent No. 7,000,219. 14 Feb. 2006.
- [83] Whittaker, James A. "What is software testing? And why is it so hard?" IEEE software 17.1 (2000): 70-79.
- [84] Welbourne, Theresa M., Diane E. Johnson, and Amir Erez. "The role-based performance scale: Validity analysis of a theory-based measure." Academy of management journal 41.5 (1998): 540-555.

## APPENDIX A: SELECTED LITERATURES

Overview of selected literatures

Database	No.	Title	Author(s)	Publish year	Research area	Source	Keywords	Reference
Inspec	1	A comparative study of software testing techniques.	Auju Bansal	2014	Software testing	Journal	Process, techniques	[67]
	2	A walk through of software testing techniques	JaiGaur, Akshita Goyal	2016	Software testing	Conference article	Process, techniques	[1]
	3	Software testing techniques: A literature review	Muhammad Abid Jamil	2016	Software testing	Conference article	Process, techniques	[68]
	4	Software testing techniques	ShivKuamr Hasmukhrai Trivedi	2012	Software testing	Journal	Process, techniques	[69]
	5	Studying the feasibility and importance of software testing: Analysis	Dr.s.s.Riaz Ahamed	2009	Software testing	Journal	Structure, management	[29]
	6	A comparative study of manual and automated testing for industrial control software	Eduard Enoiu	2017	Software testing	Conference article	Process, techniques	[70]
	7	A software function testing method based on data flow graph	Wenjing Cao	2008	Software testing	Conference article	Process, techniques	[71]
	8	Overcoming Barriers to self-management in software teams	Nils Brede Moe	2009	Software engineering	Journal	Structure, management	[79]
	9	Organizational and customer related challenges of software testing: An empirical	Frank Philip seth	2014	Software testing	Conference article	challenges	[37]

		study in 11 software companies						
	10	Modeling software testing process using HTCPN	Lizhi Cai, Weiqin Tong	2009	Software testing	Conference article	Process, techniques	[72]
	11	A Embedded software testing process model	Huaming Qian	2009	Software testing	Conference article	Process, techniques	[73]
	12	Evaluation and measurement of software testing process quality applicable to software testing laboratory	Yang Sun	2015	Software testing	Conference article	Process, techniques	[74]
	13	Identifying a subset of TMMI practices to establish a streamlined software testing process	Kamilla Gomes Camargo	2013	Software testing	Conference article	Process, techniques	[75]
	14	Agile approach for software testing process	Rijwan Khan	2016	Software testing	Conference article	Process, techniques	[76]
Google scholar	15	What is software testing? And why is it so hard?	J.A.whittaker	2000	Software testing	Conference article	challenges	[83]
	16	Choice of organizational structure for software companies in growth stage	Lijian She	2010	Software engineering	Conference article	Structure, management	[80]
	17	The influence of organizational structure on software quality: An empirical case study	Nachippan Nagappan	2008	Software engineering	Conference article	Structure, management	[81]
	18	Software testing research: Achievements, challenges,	Antonia Bertolino	2007	Software testing	Conference article	challenges	[2]

		dreams						
	19	Analysis of black box software testing techniques: A case study	Mumtaz Ahmad Khan	2011	Software testing	Conference article	Process, techniques	[77]
	20	Graybox software testing methodology embedded software testing technique	Andre'c coulter	1999	Software testing	Conference article	Process, techniques	[78]

## **APPENDIX B: INTERVIEW QUESTIONS**

### **Interview**

#### **A. Respondents' relevant information:**

1. What is your name?
2. How many years have you been in contact with the software industry?
3. How many years you have worked as a software tester?
4. What is your role in testing department?
5. Are you involved in test management activities as well or you are mainly responsible for test development and execution?
6. What is your educational background? (University degree? Which level? Undergraduate? master? phd?)

#### **B. Company information**

1. What is the size of your organization?
2. How many developers in your organizations?
3. How many testers in your organizations?
4. What kind of products or projects you are working with?
5. What are the technologies your organizations are using? (C? C++? Java? Oc?)
6. Which development methodology your organizations are using? (Agile? Waterfall? Hybrid? Plan driven?)

#### **C. Interview find answers to our RQ1 related questions**

1. How does your organization structure their testing team?  
(Case 1: One team include several developers and several testers. They work together.)  
(Case 2: Several expertise teams, such like three develop teams and one test team. They work independently of each other.)  
(Case 3, 4, 5,)
2. How did your origination manage their test team? It is same as managing development team or not?
3. What principles does the test manager rely on to assign test tasks?
4. Does the test team completely follow the test team manager's leadership?

#### **D. Interview find answers to our RQ2 related questions**

1. When will your organization start testing? (Start of the project? Mid-term project? Development work is over? and others)
2. When will your organization end the test? (Before the project is delivered/After the project is delivered and so on)
3. How many parts of the test practice process? What are each part name? (Test plan, test design, test execute, re-test,)
4. Which part do you think is the most important part of the testing process?
5. In general, how long does your organization perform a test process?

6. What testing techniques do you use in your organization?  
(Black-box, White-box, Gray-box?)  
(Dynamic testing, static testing?)  
(Manual testing, automated testing?)  
(Functional test, nonfunctional test?)
7. What are the reasons for choosing to use these technologies?

E. Interview find answers to our RQ3 related questions

1. How do you manage the test team? (only for test team manager)
2. What are the main testing challenge in software startups encounter in initial years?
3. What do you do to avoid these challenges? (challenges from E2)
4. What do you think is the main challenge in the test team management? (experience, time, budget)
5. What do you do to avoid these challenges? (challenges from E4)
6. What do you think is the main challenge in start-up companies? (your organization: time, experience, budget? ,)
7. What do you do to avoid these challenges? (challenges from E6)
8. How do you think the importance of testing in software startups?