INVESTIGATING WHAT SKILLS AND CAPABILITIES THAT SOFTWARE TESTERS NEED: A NEW ZEALAND STUDY

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF COMPUTER AND INFORMATION SCIENCES

Supervisors
Jim Buchan

February 2018

By
Ling Liang
School of Engineering, Computer and Mathematical Sciences

Copyright

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

梁 玲

Signature of candidate

Acknowledgements

First of all, I am quite appreciate the opportunity for my master study. I spent so much precious time studying and thinking, and have made great progress both in my study and personal lives.

I would like to thank my supervisor, Jim Buchan, director of Software Engineering Research Lab (SERL), School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, for his consistent guidance, inspiration, encouragement and patience. The research is entirely new area for me, and I have to update my ways of thinking. Thanks to Jim. He guides me with great patience and intelligence. He does contribute a lot to the completion of this research. The most important thing is he cultivate me not just academic writing but also critical thinking.

I would also like to thank my husband Weili Xu, for supporting me in every endeavour throughout this great journey. He is the one who shows greatest faith in me. Thank my parents and father-in-law, they help me take care of the family, so that I can focus on the study. Also, I will never forget my two lovely boys. They give me the unending power to move on. Without the encouragement and love of my whole family, I can never get the chance to make endeavours on my study and discover what I can do.

I could not leave out my friends made in SERL, Weihua Li and Barry Dowdeswell, who help me keep a positive attitude about my life and study. I become more confident and optimistic because of you.

All the best to everybody!

Abstract

Software testing has changed a lot with the rapid developing software market in the last two decades. Many companies have to either keep up with competitors or gain a competitive advantage by the continuous deploying high quality and user-friendly products. This depends heavily on the efficiency and effectiveness of testing.

New techniques, tools and methodologies are emerging in software testing, which increases the requirements on skills and capabilities for testers and has caught the shortage of test professionals. For example, they are now expected to be involved in software development to the earliest extent due to the prevalence of agile methodologies. Traditional exploratory testing is not enough and automation test is introduced.

The industry has realised the importance of software testing, but the perception of many people on the skills and capabilities for testers is limited. In some people's cognition, testers explore defects by input and click; almost everybody, either with or without IT background, can do the testing job. It is essential to understand the roles and responsibilities of testers in contemporary software development environments and the corresponding skills and capabilities the testers need. It may benefit different communities, including employers, test practitioners or potential testers, educators and researchers who are doing similar research.

The thesis aims to investigate the most important skills and capabilities for testers and gain insights into the reasons why the skills and capabilities are more, or less, important.

To conduct the research, a systematic literature review was performed to get an overview understanding of the status of this topic. There were just a few studies on this topic. As software testing is an essential part of software development, we extended the search scope to software development team. The key concepts of the research were explained, and the current understanding on this topic from references was presented. The top important skills and capabilities and their associated importance were illustrated.

A two-pronged approach was taken to answering the research questions. Job adverts were designed to answer the research questions from employers' perspective. Those adverts reflected the most favourable skills and capabilities that employers wanted for testers. Data were collected from the most popular websites that contain job adverts for testers in New Zealand (NZ) and analysed using content analysis approaches. The whole process was conducted in a hermeneutic circle and complemented using VBA and python. The frequency of each of the skills and capabilities from the sample of adverts was used to rank the importance and identify the most important ones. Besides, a conceptual model of skills and capabilities for testers was generated to compare the importance of different sources and present the findings. The categories for roles and seniority levels were also identified. On the other hand, interviews were conducted to gather opinions from test professionals who had rich testing experience in NZ. Data were collected using both semi-structured questions as well as structured questions using the Likert scale. Semi-structured questions were designed to gain a general understanding about the most valuable skills and capabilities of testers while structured questions were used to gather interviewee' view on the importance levels of each of the skills and capabilities in the proposed model and speak out loud the reasons for the importance. Thematic analysis was performed for semi-structured questions while frequency distributions and median of each skill or capability were used to analysis the importance levels.

In conclusion, this study adds knowledge to the skills and capabilities for testers and may benefit different stakeholder categories: employers, testers, educators and researchers. Further research can be conducted to gain broader and more comprehensive understanding of this area.

Publications

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

A software tester is an essential role in most software development teams, often associated with the responsibility of upholding application quality. Good testers can save the bottom line of an enterprise by discovering the defects in different artefacts at the early stages of development, or even preventing defects before they emerge.

The current fast-growing software market has led to high demand for capable software professionals, including software testers (Holtkamp, Jokinen & Pawlowski, 2015). To meet this high demand, a workforce of testers is needed that have the Skills and Capabilities (S&Cs) to do the work required of a tester in a modern software development environment. This begs the question: what are the characteristics of a capable tester in today's software development industry? Traditionally, software testers have been responsible for designing and implementing test plans and test cases to find bugs in software that developers have written (Itkonen, Mäntylä & Lassenius, 2009). The S&Cs to this work have included the ability to use test-case design techniques (Itkonen et al., 2009). The S&Cs needed to be a good tester is a moving target, however. In the last two decades, software organisations have adopted new software development techniques, tools and practices to either keep up with competitors or gain competitive advantage in an ever-changing business environment. For example, the prevalence of Agile methodologies has revolutionised much of the software industry with new ways of working and values in the software development process. Concepts such as Continuous Integration and Continuous Delivery (CI/CD) are becoming widely accepted and applied, and testers in this environment need test automation S&Cs to implement CI/CD. The role and responsibilities of a tester are evolving because of changes in software process, and emergence of new technologies that can support testing. In addition, new technologies such as Artificial Intelligence (AI) and the Blockchain are becoming part of software applications and these need to be tested, requiring new testing S&Cs. New testers need to be trained so they are competent in these new processes, methods and tools, and existing testers need to adapt and up-skill to meet the needs of modern software development. Without suitable training, re-training and recruitment a misalignment may develop between the skills of the workforce and the S&Cs needed of a tester to competently do their work in a contemporary software development environment. It is therefore important to understand the roles and responsibilities of testers in contemporary software development environments and the corresponding S&Cs the testers need to develop to competently carry out this work.

An understanding of the skills and competencies needed by contemporary software testers for different roles and different levels of seniority will be useful for those recruiting testers to design suitable job descriptions or adverts, for designers of courses for training testers to ensure development of appropriate S&Cs, and for testers themselves to guide their professional development. It is also important that there is a shared understanding of what the S&Cs of a modern tester are between these stakeholder groups so that their expectations are aligned.

Although the importance of testing and testers are widely recognised by the software community, little research has focused on the competencies that are expected of modern software testing professionals.

This thesis aims to identify the most important S&Cs expected of testers in the current industry context in New Zealand (NZ). The main sources of data include recent NZ job advertisements and interviews with practising software testers in NZ. The S&Cs expected for different testing roles and levels are identified, as well as their associated importance based on the frequency of their occurrence. The testing tasks (i.e. testing work) associated with different S&Cs is also identified. The findings from the job adverts and the interviews are compared and contrasted with each other as well as research from existing literature.

1.2 Motivation

This section describes in more detail the motivating reasons for gaining a deeper understanding of the S&Cs of testers in modern software development.

1. Software testing has been changing fast. There are some studies investigating the skills of general testers and how these skills are used in testing activities (Iivonen, 2009; Deak, 2014b). However, software testing has been changing due to the rapid development of the software industry, especially the prevalence of agile practices. So it is still important to understand the generic skills needed by testers in any software testing role in the current, ever-changing context.

2. The research is designed to support the growth of the professional software testing community. This can be done from different aspects: to bridge the gap between employers and people finding a testing job; to help educators design practical courses based on data of the most important S&Cs for testers. So, the evidence and findings of this thesis may benefit different stakeholder categories, including employers, testers, educators and researchers. This is discussed in more detail in

the next section.

3. There is not much academic research on the expected S&Cs of testers. No comprehensive framework of S&Cs are found customised for testers, although the importance of testers in software development is realised and widely accepted.

4. It is also necessary to do some research on different roles. New roles, such as agile tester, automation tester and security tester, are emerging and the most important testing skills rely much on the role of testers (ISTQB, 2016). No systematic research or investigation is found on studying the required S&Cs of specific testing roles. This is even more important for test practitioners than the general skills expected of all testers because this can enable practitioners to perform well in the specific testing role.

5. There is little literature discussing seniority levels and the career path for testers. The research on S&Cs of testers in different levels and the context or career states of testers in the real workplace can help test professionals benchmark themselves. New IT graduates can also benefit from the knowledge on seniority levels of testers.

Investigations can be conducted to discover the misalignment between the expectations of employers and the reality of testers. On one hand, testers have more responsibilities and different skill requirements than in the past so it is important to identify the S&Cs needed for testers and then find reliable approaches to develop those skills. On the other hand, employers should take care to pass the appropriate information to job seekers.

In summary, this research may benefit different communities, including employers, testers, educators and researchers, by gaining deeper insights into the expected competencies for different software testing roles, as well as the reasons behind their importance.

1.3 Research Aims

The main aim of this study can be defined as:

to identify the expected S&Cs for testers in software testing in NZ, and gain an insight into their perceptions of different testing role and seniority level.

To achieve such an aim, reviews of existing studies were performed. Key concepts include "*skills and capabilities*", "*software testers*", and "*expected*". Job adverts were analysed to gather the expectation of employers on the proposed topic. In addition, interviews were conducted not only to discover the names of those S&Cs for testers, but also to collect evidence of how they function under certain testing circumstances and why they are important for testers.

The scope of the research was restricted to the NZ market, although may be extended in future research.

1.4 Research Objectives

The primary aim of this thesis is to get a shared understanding of what S&Cs are expected for software testers in the industry of NZ. This can be refined into several objectives.

This research is expected to contribute positively to the ST industry. By considering who really wants to gain an understanding of this industry, who can potentially benefit from our research, and what kind of information they are interested in, several potential beneficiaries, including employers, testers, educators and researcher, are identified.

The objectives of this research relate to the possible uses of the knowledge of the important tester S&Cs and are described as follows:

- To help employers design or refine professional job adverts for specific testing positions and recruit appropriate or “well fitting” testers (Kanij, Merkel & Grundy, 2015). It is a challenge to recruit the appropriate testers for software organisations. There are several different roles in software testing, including several testing specialists. Each of these roles has some specific requirements on S&Cs. Employers have to design their job adverts to attract more potential applicants, especially the suitable ones. The proposed research will help recruiters to confirm what S&Cs are most valuable for their vacancy. They can quickly work out what level of testing roles they want and what S&Cs should be included in their job adverts.

- To help testing practitioners or "future" testers to “benchmark” themselves and sharp their skills to deliver high-quality software on an on-going basis (Rooksby, Rouncefield & Sommerville, 2009; Yu, Xu, Liu & Sheng, 2012). Testers can check the list of required S&Cs needed for a role, and reflect their skill gaps between the current testing role and the target role. Then, they can identify which skills are the weakest and need to be improved most (Acuna, Juristo & Moreno, 2006). This may increase their career choices.

- To help educators design practical tertiary-level courses on software testing and cultivate qualified testing practitioners needed in the workplace. They could develop practical courses based on the understanding of what skills are needed for testers and the related context that testers work in real world. With the findings in this research, they are more likely to provide students with the opportunity to practice the necessary important S&Cs need in the workplace.

- To provide researchers inspiration on performing a similar research. For example, a proposed framework on the S&Cs can be reused by other researchers in the area. Besides, other researchers may be inspired on how to perform a similar study.

1.5 Research Questions

To focus on the research, the Research Questions (RQs) have been established to keep align with the research objectives.

In this study, we will define the different testing roles, identify their responsibilities and associated skills. The whole investigation will be based on the context of New Zealand.

The research questions can be defined as follows:

RQ1: *What are the most important generic skills and capabilities needed by any tester, and why?*

RQ2: *What are the most important skills and capabilities that are specific for different testing roles, and why?*

RQ3: *What are the most important skills and capabilities of testers in different levels, and why?*

1.6 Research Approaches and Design

In this thesis, several research methods are used, including literature review (LR), job adverts analysis, and semi-structured interviews, and different data analysis methods.

Through LR, two possible reasons for the shortage of test professionals are found in the industry. One is the recruiters or people who write the job descriptions are not test professionals and have a biased understanding of what kind of people they are exactly looking for. Another reason is that the requirements on S&Cs for testers is gradually increasing.

The proposed research investigates the most important S&Cs from perspectives of both employers and test professionals. Then, the findings from those investigations can help the potential readers, including employers, test practitioners, educators and

Table 1.1: Implementation of the Research

	Part 1	Part 2
Data Generate Approaches	Job Adverts	Semi-structured and Structured Interviews
Data Analysis Approaches	Content Analysis Hermeneutics Circle	Thematic Analysis

researchers, reach the objectives mentioned above.

The research is performed in two ways: analysing documents (job adverts) and data from the interview. Table 1.1 shows the main research approaches employed in this thesis.

1.7 Contribution

There are several contributions in this thesis:

- It adds knowledge to the S&Cs for software testers and contributes positively to the language of this topic. A conceptual model of the S&Cs for testers was built to organise the interviews and final findings, which may provide a structure for future research. Besides, categories of testing roles and seniority levels were identified which can be reused by other researchers.
- It can deepen the understanding of the top generic S&Cs for any tester, as well as those for different roles and seniority levels.
- It can add value to four different stakeholder categories, including employers who are recruiting testers, test practitioners or "future" tester who want to find a testing job, educators who are planning a testing course and researchers who are researching the proposed area.

The finding of this study can provide deep insight into the career blueprint for software testers. It is expected to answer the questions, such as how to become a good tester, what should be done to shift to another role.

1.8 Thesis Outline

This thesis is organised as follows:

Chapter 1 - Introduction: introduces the context of the research, including the motivation, objectives, design and contribution.

Chapter 2 - Literature review: elaborates the research context, research process, and current understanding on the RQs based on a systematic literature review.

Chapter 3 - Research design and implementation: discusses the design and implementation process of this research.

Chapter 4 - Analysis: presents the roles, seniority levels and a customised conceptual model of the S&Cs for testers; analyses the findings structured by RQs.

Chapter 5 - Conclusions: summarises the main findings and answers for the research questions, illustrates the limitations of this research, and potential future work.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides a comprehensive picture of the current understanding on the topic using a Systematic Literature Review.

Section 2.2 introduces the approaches for collecting the related literature, including the search process and strings.

In Section 2.3, the key concepts of the study, i.e., *software testers*, *skill* and *capability*, are explored and defined; their importance is discussed respectively. It is essential to clarify the research aims and main concepts before reviewing literature. This can not only help the main researcher focus on the research questions and objectives, but also help potential readers to understand the context and findings of this research. In this thesis, the main aim is to establish the most important *skills and capabilities* of *software testers*.

In Section 2.4, the existing literature is reviewed and the findings are presented. A general understanding can be gained about the S&Cs for testers found in current studies. This helps to generate the research questions and develop the conceptual skill framework for testers.

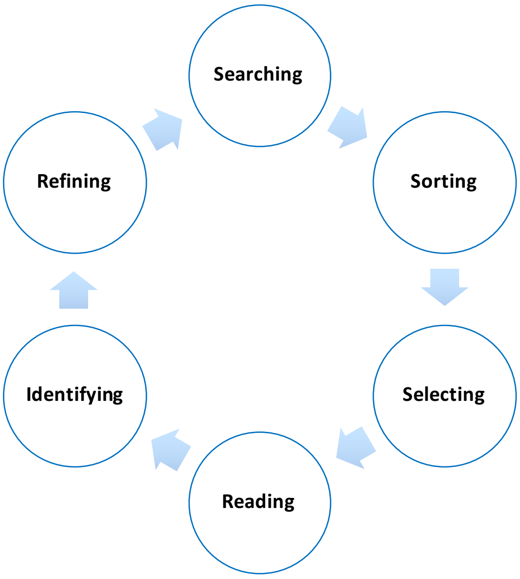


Figure 2.1: The Searching Circle of Literature

In Section 2.5, the proposed research is discussed, which simply describes the motivations and value for doing this research.

2.2 Literature Search Approaches

2.2.1 The Search Process of Literature

Finding existing literature on the research topic is the first step in a Literature Review (LR). This is one of the most popular ways to gain the current understanding of the topic. A well-designed search can help discover sufficient valuable documents for the research. The design of search process in this thesis is illustrated in Figure 2.1.

The stages include:

- Generating initial search strings from the keywords of this research;
- Sorting the results by dates and relevance;
- Selecting the filtering conditions, such as subject area, keywords and paper source;

- Looking through selected articles to gain a deep understanding of the study area, and paying attention to useful references;
- Identifying central terms and useful references for further searching;
- Refining search strings and searching new articles found in existing articles.

The search process may happen at any stage of the research process. It is a process of continuous exploration. The circle may stop when the literature becomes comprehensive enough to answer the research questions and no new valuable articles can be identified.

2.2.2 The Selection of Online Databases

There are several types of frequently used literature, including academic literature, professional literature, and ’grey literature’ (Hammond & Wellington, 2013). In this research, both peer-reviewed journal articles and conference papers are considered during LR. There are some reasons for this. First, these web-based documents, including conference papers, have increasing importance in revealing the current status of the field (Hammond & Wellington, 2013). Second, the topic is more related to industry practice and it is different from studies of creating a new technique. So a flexible review will be undertaken. This will be more discriminating.

To get enough information in the research field, several databases, such as IEEE, ACM, and Scopus, were used. The general search has been done on the Scopus because it has the links to literature in the other databases. Google Scholar is also been used to get the link or document of related literature when the article name is known. It performs well in finding the reference cited in some articles.

2.2.3 Inclusion and Exclusion Criteria

The inclusion/exclusion criteria can not only help readers understand the scope of the review, but also help the main researcher focus on the research area and prevent bias from selection of studies.

The review consists of three main parts: the definitions of key items; current understanding on S&Cs for testers and related frameworks; S&Cs emphasised on existing testing courses.

Generic criteria of inclusion can be described as follows:

- Publication year is >= 2007. The research aims to discover the S&Cs for testers in a modern software development environment.
- Language: English.
- Type of publication: published in an academic conference proceedings or a journal

As this research aimed to identify the modern requirements of S&Cs needed for software testers, initial criteria was set to refine the findings.

- The findings should contain the S&Cs needed for software testers.
- Subject of area were limited to "Engineering" and "Computer Science" because software testing is an integrated part of software engineering.
- The research gives priority to those studies with the highest frequency of citations and newest publish date because they could reflect the most recent and authorised findings respectively to some extent.

There exclusion criteria were basically the opposite of the inclusion criteria.

2.2.4 Search Strings

At the beginning of the research, key words "skill" and "tester" was used to get an general understanding in existing studies of the research area and identify synonyms of the key items for further comprehensive search.

During the research, three main search strings were determined. The first one includes the key concepts of the research topic.

Search string 1:

("skill*" OR "capabilit*" OR "competenc*" OR "characteri*" OR "good" OR "high performance") AND ("tester" OR "testing" OR "software engineer" OR "software professional" OR "software development")

The first string consist of two main parts. The first part contains the regular expressions related to skills and capabilities; the second part was designed to find the studies about the professionals in software testing. The first string was not limited to software testing or software tester to avoid missing valuable documents. As software tester is an important role belonging to software engineer, so, "software engineer", "software professional", and "software development" was added to extend the results. The initial results includes 133,671 documents.

A few rules were set to narrow and refine the range of results:

- Search keywords "technical skill" and "soft skill" were chosen.
- Search results were limited to publish years 2007-2017. This filter was set to get an overview of the newest findings on the research area.
- Subject areas "Engineering" and "Computer Science", were selected.
- Document types were limited to "Article" and "Conference Paper" to make the documents found more academic.

- The results were listed in order of the newest date by default and then, "Cited by (highest)" were chosen to get the potential most useful documents in the relevant field.

Then, similar rules were applied to other search strings.

Search string 2:

"technical skill*" OR "hard skill*"

Search string 3:

"soft skill*"

These two search results were combined with Search string 1 using the function provided by Scopus. 106 documents were identified by associating string 1 and 2, while 67 documents were obtained using string 1 and 3.

Search string 4:

"pedagog*" OR "teaching" OR "curriculum" OR "course" AND "software testing".

Similarly, some keywords were identified, including "software testing", "teaching", and "Curricula". As a result, 26 documents were found after the filter conditions which seem to be most close to the topic.

Studies were collected and determined through the iterative process: when a valid study was found, all its reference were checked to find more relative studies that were published earlier.

References in the searched useful articles were checked for further judgement.

The searching process could be summarised as follows: use simple search to establish the synonyms of key words on the research area; narrow the documents results by pre-defined inclusion/exclusion criteria; search more related studies from the references of existing literature. An initial evaluation was made by information provided by the title and abstract. Then, potential useful articles were downloaded and imported into Mendeley[1], a free reference manager. Potential useful documents were eventually marked by the main researcher by reading through the content, while useful points and references were extracted and summarised.

During the whole process, other search strings were applied. For instances, different keywords, such as "what is a skill", "skills defined", "capabilities defined", were used to search the definition and meaning of concepts.

In summary, all the reviews on literature focus on the S&Cs needed for testers, why they are important and how to obtain or cultivate them. Search string 1 was set to be quite broad to avoid missing important studies. This could be refined in some way to improve effectiveness.

The results for the search string were validated by comparing the returned results with the references lists from several of the significant, highly cited articles and checking there were no major gaps in the results.

2.3 Key Concepts

Before starting research, the related key concepts should be defined and the scope of study should be refined. This provides the context of the research field.

The first concepts are Software Testing (ST) and Software Tester. Questions can be asked, such as: "what" is ST? "who" is the software systems tested by (Wong et al., 2011)? Why are they so important?

2.3.1 The Context of the Topic

Software testing (ST) and software tester fail to attract as much attention as other roles in the development team, such as software developers and analysts (Iivonen, 2009). Three negative human factors are identified (Fernández-Sanz, Villalba, Hilera &

[1] https://www.mendeley.com/

Lacuesta, 2009): instability of testers' positions, lack of attractiveness of testing and poor career development for testers. Many people's perception about ST stays at that of 20 years ago. In their opinions, ST is not a challenging job because less technical skills are needed in testing, compared with software development. It can be done by simply clicking the buttons and checking the output of software systems with expected results. Practitioners from different educational backgrounds, not just those with IT background, can do the manual testing job well and become professional after some years' experience. Also, there are other demotivated factors, including lack of influence, bad relationship with co-workers, time pressure, etc. (Deak, 2014a; Beecham, Baddoo, Hall, Robinson & Sharp, 2008). Testers are confronted with tricky situations, dealing with different stakeholders and developers (Yu et al., 2012).

However, software testing has changed a lot with the rapid development in the market. For example, agile methodologies have introduced significant technical changes in software development in the last two decades. Agile has shown better performance, such as reducing cost and time to market, compared to traditional methodologies, such as Waterfall (Papadopoulos, 2015). Many Software Development (SD) companies find that they can no longer achieve a competitive advantage just by lower prices and increased productivity in the global software context (Sengupta, Chandra & Sinha, 2006). They have to deliver high-quality software products to the market on-time. This relies heavily on the efficiency and effectiveness of testing, or the higher requirements of the S&Cs of testers (Rivera-Ibarra, Rodríguez-Jacobo & Serrano-Vargas, 2010). So CI/CD become prevalent and testers are expected to get involved as early as possible. The agile development process changed the traditional way of doing testing.

Although the status and importance of software testers in software engineering are gradually realised by the industry (Bertolino, 2007), the lack of qualified testers is still the obstacle of software engineering (Bin & Shiming, 2013). It is estimated that more than half of the development budgets are spent on testing (Day, 2014; Krutz, Malachowsky & Reichlmayr, 2014) and one-third of the cost spent on defects could be reduced (Wong et al., 2011). In contrast, the number of specialised professionals in the test industry is far less than the demand (Valle, Barbosa & Maldonado, 2015).

Employers find it difficult to recruit enough appropriate testers for several reasons. First, the expectations placed on highly-skilled software testers is increasing due to the growing complexity of software development (Astigarraga, Dow, Lara, Prewitt & Ward, 2010). For instance, the ability to do automation testing is expected for many testers nowadays. It has increasing importance in software engineering and is considered as the core of agile testing (Collins & de Lucena, 2012). However, the survey conducted in Canada indicated the dominant method of ST is still manual testing, not automated testing recommended in agile development environments (Garousi & Zhi, 2013). The main reason for this phenomenon is the insufficient number of automation testers and the skill-set of existing testers. Besides, Collins and Lucena (2012) pointed out that enterprises find it difficult to recruit people with the right tester mindset and right skill-set (Collins & de Lucena, 2012): Testers do not have a programming background while developers do not have experience in testing. It is quite common that in some small companies, developers have to test their own code, such as unit test and integration testing. But due to the time-frame, only positive cases, or "happy" paths are tested. Meanwhile, many developers are not likely to admit the defects in their code. These are the main reasons why testers are needed to do software testing.

This is one of the main *motivations* of this research, to investigate what S&Cs are now most expected for all software testers and any specific test roles or specialists.

2.3.2 Software Testing and Software Testers

Software Testing

ST, a significant determinant of software quality (Kanij, Merkel & Grundy, 2013), has a wide variety of definitions in the literature (Nouman, Pervez, Hasan & Saghar, 2016).

“Software testing is the observation of the execution of software based systems in order to verify that the system behaves as expected and to identify defects in the system under test.” (Whyte & Mulder, 2011, pp. 255)

Software testing is the activity of identifying defects to evaluate and improve the quality of the product (Matturro, 2013). Software testing is the process of dynamic verifying and validating the behaviour of a product on a finite set of test cases, against the expected behaviour (De Souza, Falbo & Vijaykumar, 2015).

"Software testing is the activity performed for evaluating product quality, and for improving it, by identifying defects and problems."(Matturro, 2013, pp. 135)

“Software Testing consists of dynamic verification and validation of the behaviour of a program on a finite set of test cases, against the expected behaviour." (De Souza et al., 2015, pp. 379)

International Software Testing Qualifications Board (ISTQB)[2] identified four primary objectives of ST: finding defects of all artefacts as early as possible, gaining confidence from software quality, providing information for decision-making and preventing defects from emerging (Muller & Friedenberg, 2011). It has a close relationship with Quality Assurance (QA).

In this thesis, ST is defined as: the process to improve software quality by finding and preventing defects in related artefacts using appropriate techniques, resources and tools.

ST is a crucial, professional activity that should be kept in alignment with other software engineering activities, such as architecture and development (Weyuker, Ostrand, Brophy & Prasad, 2000). Effective software testing is essential to the quality of software systems, which can bring the development team confidence in the different

[2] www.istqb.org

quality attributes, such as reliability, dependability and performance (Weyuker et al., 2000).

As software applications are an integral part of human life now, software testing has demonstrated its importance in the development of high quality software. Defects of software products are inevitable for some reasons, such as limited time or resources, the complexity of code, ever-updating technologies, and system interactions (Muller & Friedenberg, 2011). Failures caused by software can lead to different problems, such as losing money, time and even causing injury or death. Software testing can help to reduce or migrate both product and project risks, which is a primary practical approach to ensure the software quality (Wong et al., 2011).

Software Testers

So, what are testers? Simply speaking, a software tester is a person doing ST to assure the product quality. It has many similar names.

Kanij, Merkel, and Grundy (2015) defined a software tester as the person who helps the developers to increase the quality of a software product using different tools and techniques. Cerqueira (2016) emphasised that testers are not just the “filters” of software quality anymore but the main "guard" of quality. They are actively involved in the development process from beginning aiming to prevent bugs from emerging at all.

The definition of software tester is given according to the objectives of ST (Muller & Friedenberg, 2011): A software tester is a person who is involved in testing activities to find and prevent bugs.

When developers test their code, they mainly focus on if the software achieves business requirements under optimal conditions.

Software testing has changed a lot in the last two decades, so are the roles of software testers. General roles of software testers include Test Manager (TM), Test Lead (TL), Test Analyst (TA). Technical test roles, such as Integration Tester, Performance Tester and Automation Tester, belongs to test specialist. There is a particular role: Agile Tester.

An agile tester is a person doing testing activities in an agile context. This person always is the most experienced engineer in ST in the development team (Deak, 2014a). In the context, the quality of a software product is the responsibility of the whole development team, rather than that of specific tester or developer. People are involved in ST since the beginning of the project. Unit tests are performed before coding, which may relate to methodologies including TDD.

2.3.3 What is Meant by Skill and Capability?

One of the main aims of this study is to identify the expected skills, capabilities and attitudes of all testers in the software testing industry in NZ, as well as those of different testing roles.

The Definitions of the Concepts

In many studies about this topic, the item "skill" is used directly without definition (Beer & Ramler, 2008; Yilmaz, O'Connor & Clarke, 2012), and apparently similar expressions are widely used in different references. Originally, a skill was simply defined as "an ability acquired through practice" (Crossman, 1964). But the definitions from different dictionaries reveal the complexity of this skill (Attewell, 1990). For example, the definition of skill in Collins English Dictionary is expressed as follows: "Skill is the ability coming from one's knowledge, practice, aptitude etc., to do something well; competent excellence in performance; expertness; dexterity" [3], which is based on the idea of competence or proficiency. Through this definition, skill denotes the ability to do something and can be increased or sharped through learning and practice.

The concept of capability emerged from UK in the middle of 1980s due to the rapidly increasing complexity of work and organisational competitiveness (Hase & Davis, 1999). It denotes "an integration of knowledge, skills and personal qualities used effectively

[3] http://www.dictionary.com/browse/skill

and appropriately in response to varied, familiar and unfamiliar circumstances" (Phelps, Hase & Ellis, 2005). It reflects skills and knowledge needed in the workplace (Hase & Davis, 1999). Capable people "know how to learn, are creative, have a high degree of self-efficacy, can apply competencies in novel as well as familiar situations and can work well with others" (Hase & Davis, 1999). Adaptability and being creative are typical examples (Phelps et al., 2005).

The capability is a holistic attribute which includes the ability to obtain knowledge and skills (Hase & Davis, 1999). This means one can learn particular skills quickly and make things work as expected with capabilities[4].

So the relationship between capabilities and skills are:

Capabilities are abilities or qualities to acquire the needed skills.

In this thesis, efforts are not made on specific categories of S&Cs separately.

The Similarity and Difference of Other Concepts in Reference

Competency and competence are two widely used expressions for skills (Beer & Ramler, 2008; Yilmaz et al., 2012; Kanij, Merkel & Grundy, 2014; Holtkamp et al., 2015). The literature did not explicitly distinguish them.

The definition of competency varies. Competency was stated as "individual and measurable skills according to standards of competence" (Phelps et al., 2005). It emphasises the ability to competently perform tasks in a job (Velde, 1999; Mulcahy, 2000). Competency is widely used to denote the requirements of different roles in a job(Matturro, 2013; Robles, 2012). Competency refers to "a collection of skills, abilities, and attitudes to finish tasks in a given working context", while competence relates to a specific skill, ability, or attitude of an individual (Winterton, 2009; Pawlowski & Holtkamp, 2012).

However, skill is "not only synonymous with competence, but also expertise and mastery" (Attewell, 1990).

[4] exilelifestyle.com/skill-capability/

Hard Skills and Soft Skills

Basically, skills are considered as those abilities learnt from knowledge, training or work experience. In many studies, skills are simply classified into hard (technical) and soft skills (Holtkamp et al., 2015). This is widely used in the jobs adverts in software engineering industry to organise the skill requirements (Ahmed, Capretz, Bouktif & Campbell, 2013).

Hard skills, also well known as "technical skills" (Wikle & Fagin, 2015), "are the technical expertise and knowledge needed for a job" (Robles, 2012, pp. 453). Popular hard skills in the software testing include testing domain or context knowledge, programming knowledge, testing tools and techniques, etc. This can always be demonstrated by training, education and certification, especially practical testing experience. Here, testing experience can be defined as "practical knowledge developed by direct observation of or participation in testing activities" (Beer & Ramler, 2008), which are also emphasised by several researchers (Kaner, Bach & Pettichord, 2002; Matturro, 2013).

Soft skills are complementary to one's technical skills. Soft skills are rated as the first in importance for entry-level success in a job (Wilhelm, 2004). As defined in (Lacher et al., 2015), Non-Technical Skills consists of cognitive skills, personal resources and social skills which are conducive to job tasks. Robles (2012) argued that soft skills are character traits, attitudes, and behaviours, based on the definition of Collins English Dictionary[5]. They consist of interpersonal (people) skills and personal (career) attributes.

Soft skills required for graduates was well-known as professional skills (De La Harpe, Radloff & Wyber, 2000). Professional skills, or employ-ability skills, was widely adopted in engineering education (Shuman, Besterfield-Sacre & McGourty, 2005). According to the research of Shuman et al. (2005), professional skills could be divided

[5] http://www.dictionary.com/browse/soft-skills?s=ts

Table 2.1: Equivalent Items of Hard and Soft Skills

Skill Category	Equivalent Items
Hard Skills	technical skills, abilities or competencies to complete a specific task, expertise acquired from education, training (Wikle & Fagin, 2015); technical expertise;
Soft Skills	non-technical skills, people skills, social skills, generic competencies, human factors (Matturro, Raschetti & Fontán, 2015); emotional intelligence (Ahmed, Capretz, Bouktif & Campbell, 2012); characteristics (Deak, 2014b); traits,behaviours, and attitudes that drive one's behaviours (Robles, 2012);

into process skills (such as communication skills and teamwork) and awareness skills (such as the ability to do lifelong learning).

Besides, it is also important to clarify the word "expected" because this topic is to investigate the expected S&Cs from different perspectives: employers and test professionals. According to an online dictionary, the term "expected" is defined as "supposed to have or required to fulfil an obligation or complete a task" [6]. In this thesis, "expected" is limited to "required to fulfil the responsibilities or tasks of testing roles".

2.4 Current Understanding of Skills and Capabilities

Most existing studies on software testing focus on developing cutting-edge tools and techniques for different types of testing (Bertolino, 2007), or current practices in SE enterprises (Causevic, Sundmark & Punnekkat, 2010; Garousi, Coşkunçay, Betin-Can & Demirörs, 2015). They seem to be more practical and useful for the software industry. But all the testing activities are decided and performed by humans. So, it is necessary to investigate human factors in software testing. After a systematic search on S&Cs of software testers, a few articles were found. Then the search was extended to software professionals because a tester is an important role for a software engineer. In this

[6] https://en.oxforddictionaries.com/definition/expect

Table 2.2: Multi-dimensional Framework of Competence

	Occupational	Personal
Conceptual	Cognitive competence	Meta competence
Operational	Functional competence	Social competence

section, the existing frameworks of skills and competencies are presented while the current findings on the most important skills and characteristics of testers are analysed and synthesised.

2.4.1 Related Frameworks on the Research Area

To develop the customised frameworks of S&Cs for testers, it is meaning to get an insight into the existing frameworks on the research area.

The generic framework for competences provided us theoretical support for later classifications of S&Cs found by different data sources, which helped the main researcher design the conceptual model for testers. Besides, some skills and their associated categories could be referenced, although the frameworks designed for software engineers were not completely match the skill set needed for software testers.

A Multi-dimensional Framework of Competences

Through the taxonomy of competence (Le Deist & Winterton, 2005), a competence can be categorised into four groups: Cognitive Competence (know-what and know-why), Functional Competence (know how in occupational area), Social Competence (know how to behave), and Meta-competence (cope with uncertainty, as well as learning and reflection).

Fig 2.2 depicts the holistic model of competence, which inspired us to build our framework.

A Theoretical Framework of Skills

A commonly used theoretical framework of skills is depicted in Figure 2.3 (Ahmed

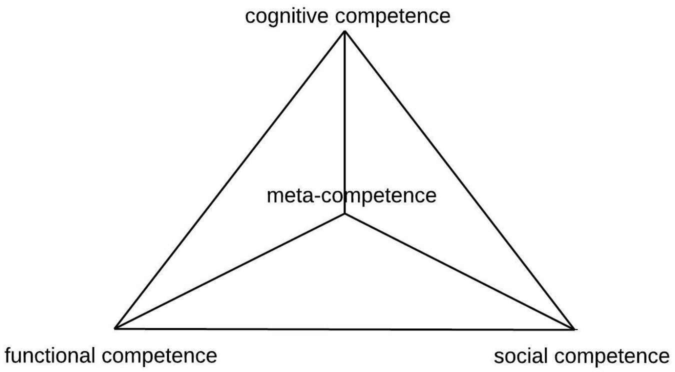


Figure 2.2: Holistic Model of Competence

et al., 2013; Mtsweni, Hö & Poll, 2016). It indicates how skills influence job performance. This model shows how the two kinds of skill requirements in a software development context influence job performance. Technical skills contribute positively to performing tasks and activities, while soft skills are mainly used for social requirements in the community, especially communication and interaction between team members or stakeholders on the same project. They affect the job performance through impact on work behaviours.

Software Engineering Body of Skills

Software Engineering Body of Skills (SWEBOS) (Sedelmaier & Landes, 2014) proposed a mixed set of technical and non-technical competencies needed in complex software development. It is "the basis for competency orientated teaching and learning". 29 competencies under seven competency categories are elaborated in the body of competencies (see Fig 2.4).

The main feature of this skill set is all the skills are described precisely and customised for software engineering. But the names of skills are too precise and are not easy to spread out.

A Competency Framework for Software Engineers

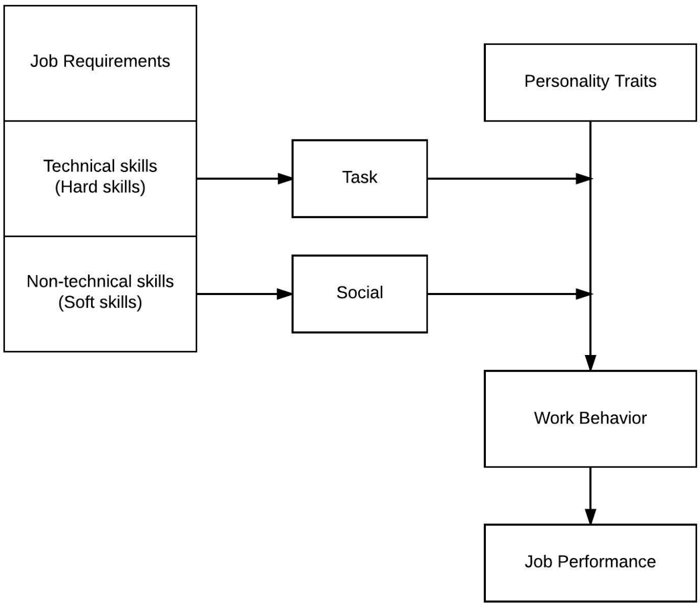


Figure 2.3: Theoretical Framework

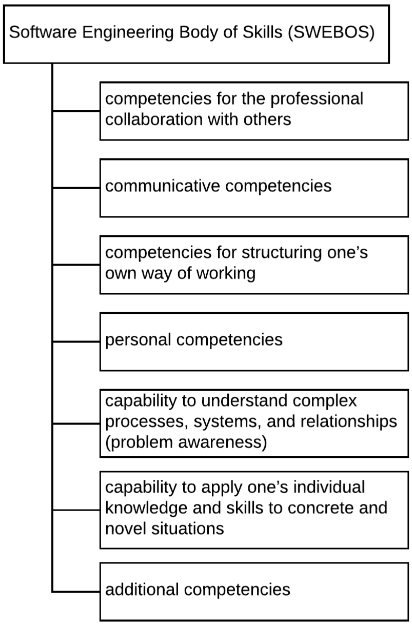

Software Engineering Body of Skills (SWEBOS)
competencies for the professional collaboration with others
communicative competencies
competencies for structuring one's own way of working
personal competencies
capability to understand complex processes, systems, and relationships (problem awareness)
capability to apply one's individual knowledge and skills to concrete and novel situations
additional competencies

Figure 2.4: Software Engineering Body of Skills (SWEBOS)

This is a competency framework generated by grounded theory(Rivera-Ibarra et al., 2010). It used to evaluate software engineers, including software testers. This article defined related knowledge (to know how to do), skills (to be able to do) and behaviour/attitudes (to want to do). The framework consists of twelve technical skills, 18 items of social skills and 24 personal qualities. All these skills are collected from two sources. Skills are initially obtained through field observations, interviews and questionnaires, and then filtered by self-defined protocols.

It is a relatively comprehensive competency collection, which is adapted by other researchers (Holtkamp et al., 2015). It provides a good way to design the competency framework. But this framework is designed for software engineers, rather than software testers. For example, many of the competencies listed "Technical knowledge" are not associated with testing.

Non-Technical Skills Inventory

The skill typology of software professionals (Lacher et al., 2015) is extracted from a systematic literature review and validated by both industry and academic experts afterwards, which provides some information to build the conceptual skill framework for testers. It includes twelve most-frequent soft skills, including teamwork, oral communication skill, and attention-to-detail, etc. This behaviour-based NT skills taxonomy is generated based on the 408 observable performance behaviours of each skill.

2.4.2 Summary of Studies on Skills and Capabilities of Software Testers

The influence of human factors on the effectiveness of testing has been emphasised by some researchers (Beer & Ramler, 2008; Shoaib, Nadeem & Akbar, 2009; Deak, 2014b). Human influence and experience are critical factors when performing test tasks (Briand & Labiche, 2004). This argument is strengthened by later empirical studies (Rooksby et al., 2009; Shah & Harrold, 2010), which contribute positively to the development of ST practices.

Knowledge and experience are critical job requirements in recruitment, especially in small companies. Organisations failing to recruit experienced testers might find it hard to develop high-quality products, especially in the context of a new or complex product or technology (Deak, Stålhane & Sindre, 2016). It is shown that experienced testers can successfully apply a wide range of practical testing methods and techniques in the context of a software project (Vegas, Juristo & Basili, 2006). During the ethnographic study of a small SE company with volatile requirements, the researchers found the importance of experience in both domain and technique in integration and acceptance testing (Martin, Rooksby, Rouncefield & Sommerville, 2007). Then, the value of experience in testing, including test automation and management, and experience in product domain was identified by the multiple-case study (Beer & Ramler, 2008), which enhanced the conclusion in an earlier study (Martin et al., 2007). The qualitative observation study performed by (Itkonen et al., 2009) verified the importance of domain knowledge in a manual testing process, especially when facing incomplete and ambiguous specifications. Then Iivonen et al. (2010) also pointed out the importance of domain knowledge and experience in testing techniques. They seem more useful than skills in designing test cases and plans. This finding is similar to that in (Kettunen, Kasurinen, Taipale & Smolander, 2010). Testers can hardly complete the testing tasks well without related domain knowledge. In the research (Kannan & Vivekanandan, 2012), the authors also emphasise the importance of functional knowledge and experience, using correlation and regression analysis. Besides, diversity in experience is considered as a key factor in building a strong team (Kanij, Merkel & Grundy, 2011; Deak et al., 2016). Employers would like to recruit testers with diversity to reduce cost because they can switch roles and complete different tasks, comparing to hiring people for different roles (Deak et al., 2016).

Soft skills have attracted much attention in the last decade (Ahmed, Capretz, Bouktif & Campbell, 2012; Deak et al., 2016). In the experiment conducted by Acuna et al. (2006), a human capability-base process was proposed to identify the fitness between personal capabilities. Those capabilities are categorised into four groups: intrapersonal, organisational, interpersonal, and management. In another survey (Ahmed, Capretz & Campbell, 2012), 500 online ads were analysed, 140 of which were related to software testers. Communication skills were identified as the only high demand soft skill for testers, while three other soft skills were labelled as moderate demand. Matturro (2013) also did an investigation on software engineers, based on the classified ads published by newspaper in Uruguay and database maintained by the Graduate Office of Universidad ORT Uruguay. A total of 678 software-related ads, including 60 on software testing or QA, are gathered and the 17 most popular soft skills are extracted after quantitative analysis. These skills are utilised by later research (Matturro et al., 2015). The survey conducted by Deak (2014b) ranked the 18 most valuable factors of testers using thematic-analysis. Among all the studies on soft skills of testers above, the majority marked communication skills as being the top importance.

The importance of testing tools and techniques is also mentioned in several studies. The investigation conducted by (Lee, Kang & Lee, 2012) indicated high demand for support tools and methods between development and testing. Besides, human psychology plays an important role in testing practice because “testing is about discovery” (Armour, 2005, pp.17). This conclusion is based on the dual hypotheses of knowledge discovery (Armour, 2001). Good testers have a special intuitive sense of how, where and how much to test during the process (Armour, 2005).

There are also some studies that focus on the motivation of software testing personnel (Shah, Harrold & Sinha, 2014). They show the work context of testers, which can help us understand the importance of specific skills in the context.

2.4.3 Ranking of Specific Skills and Capabilities of Testers

The studies on the implied research area are not rich. As "tester", usually called "software testing engineer", is a critical member of a development team and belongs to software engineers, the search scope is extended to software development team and software engineers. Findings related to testers were extracted.

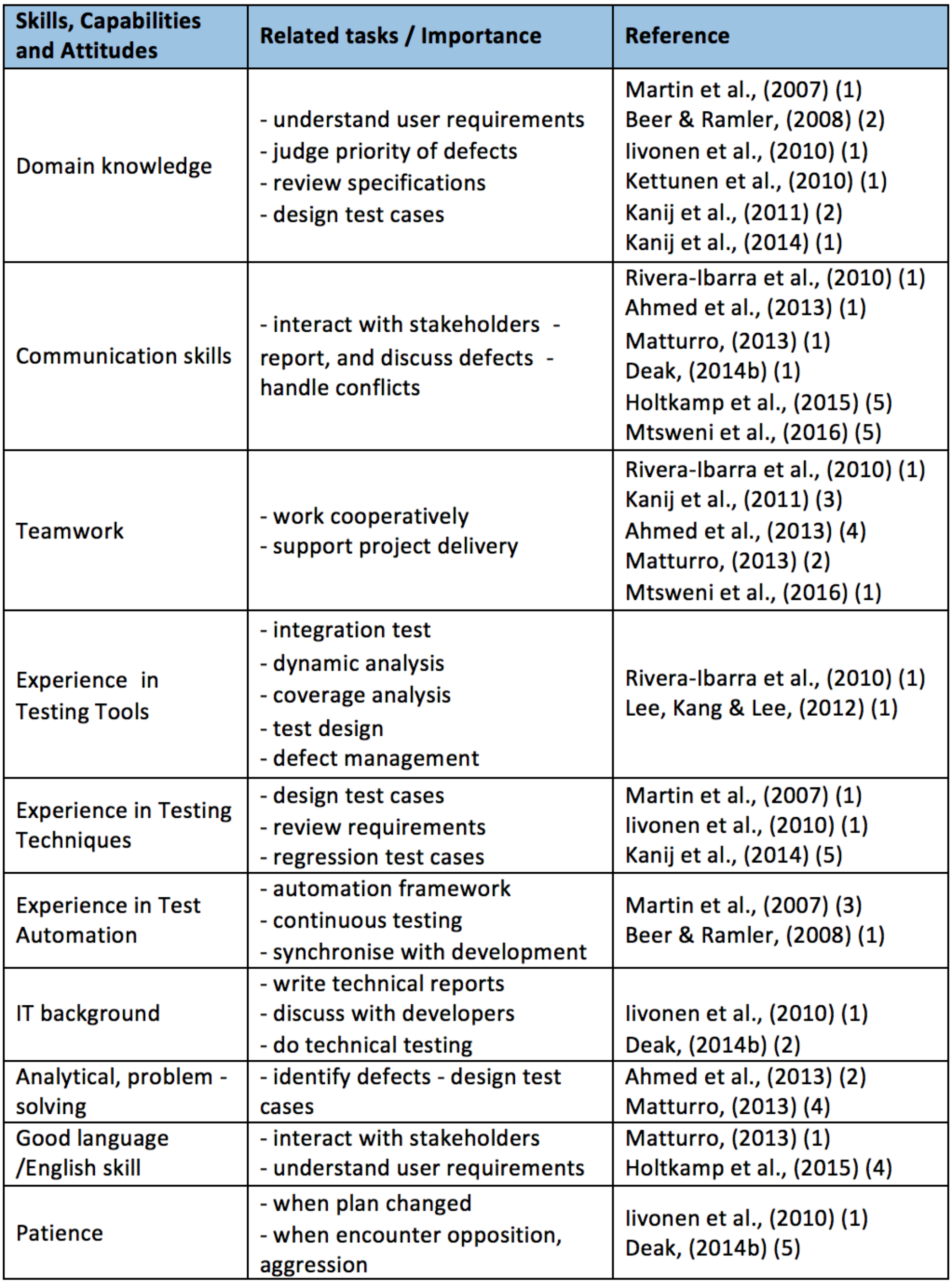
To get a clear understanding of the most important skills for testers and their importance from literature, the top-ranked skills, capabilities and attitudes are listed in Tables 2.3. All the listed items are mentioned in more than one reference.

In this table, the first column presents the names of skills, capabilities and attitudes; the second column lists their importance associated with testing tasks and activities. References to literature are provided to get an index for related studies. The ranks of specific skills, capabilities and attitudes in the studies are explicitly stated in brackets.

It is hard to quantify the importance of skills by simple rules. Different factors, such as the authority of reference, the frequency of skills mentioned in literature, and the given rank of importance in literature should be taken into consideration. In this thesis, the references are assumed to be of equivalent significance. The number of references and the ranks of skills, capabilities and attitudes on importance are employed to reflect the importance of skills highlighted in the LR.

More information can be found in Table B.1 in Appendix B. From another perspective, it illustrated the important skills or characteristics aggregated by reference. The main demographic information since 2007 is enumerated. This table also lists the research scope, where "STs" represents software testers, while "SEs" and "SDs" denotes software engineers and software development teams respectively. Table B.2 explicitly list the frequency of S&Cs found in the literature. The definitions and their importance will be elaborated in the next subsection.

Table 2.3: The Top Skills, Capabilities and Attitudes Found in Literature



Skills, Capabilities and Attitudes	Related tasks / Importance	Reference
Domain knowledge	- understand user requirements - judge priority of defects - review specifications - design test cases	Martin et al., (2007) (1) Beer & Ramler, (2008) (2) Iivonen et al., (2010) (1) Kettunen et al., (2010) (1) Kanij et al., (2011) (2) Kanij et al., (2014) (1)
Communication skills	- interact with stakeholders - report, and discuss defects - handle conflicts	Rivera-Ibarra et al., (2010) (1) Ahmed et al., (2013) (1) Matturro, (2013) (1) Deak, (2014b) (1) Holtkamp et al., (2015) (5) Mtsweni et al., (2016) (5)
Teamwork	- work cooperatively - support project delivery	Rivera-Ibarra et al., (2010) (1) Kanij et al., (2011) (3) Ahmed et al., (2013) (4) Matturro, (2013) (2) Mtsweni et al., (2016) (1)
Experience in Testing Tools	- integration test - dynamic analysis - coverage analysis - test design - defect management	Rivera-Ibarra et al., (2010) (1) Lee, Kang & Lee, (2012) (1)
Experience in Testing Techniques	- design test cases - review requirements - regression test cases	Martin et al., (2007) (1) Iivonen et al., (2010) (1) Kanij et al., (2014) (5)
Experience in Test Automation	- automation framework - continuous testing - synchronise with development	Martin et al., (2007) (3) Beer & Ramler, (2008) (1)
IT background	- write technical reports - discuss with developers - do technical testing	Iivonen et al., (2010) (1) Deak, (2014b) (2)
Analytical, problem - solving	- identify defects - design test cases	Ahmed et al., (2013) (2) Matturro, (2013) (4)
Good language /English skill	- interact with stakeholders - understand user requirements	Matturro, (2013) (1) Holtkamp et al., (2015) (4)
Patience	- when plan changed - when encounter opposition, aggression	Iivonen et al., (2010) (1) Deak, (2014b) (5)

2.4.4 The Definition and Importance of Testing-related Skills and Capabilities

There is still no unified name or definition for each skill in the literature. The definitions and importance of the most important S&Cs for testers will be provided in this section. This may contribute positively to the following research, in conducting the investigations and building a conceptual framework of S&Cs expected for testers.

Domain Knowledge

Domain knowledge refers to the realm of knowledge that one has in a particular field (Alexander & Judy, 1988). Generally speaking, both business domain knowledge and technical domain knowledge belongs to domain knowledge. However, the domain knowledge mentioned in literature stands for business or product domain knowledge.

The importance of domain knowledge is highlighted by most studies, marked as the top skills for testers. Iivonen et al. (2010) interviewed twelve test professionals and found knowledge and experiences in product domain can help testers understand the functions or features that users want to achieve and review the specifications or other artefacts. Deak (Deak, 2014b) declared the increasing sensitivity and complexity of business that testers are involved in. Many defects may be ignored if one is not familiar with the product or business. The investigations also show the importance of domain knowledge in identifying the severity and priority of defects discovered in the context. Testers with rich domain knowledge can identify the most risky features. Besides, as shown in the multiple case study (Beer & Ramler, 2008), domain knowledge can help testers design test cases, which is verified by Iivonen et al. (Iivonen et al., 2010).

In summary, testers with good domain knowledge do well in different testing tasks, such as designing test cases, identifying defects and communicating with stakeholders.

Communication Skills

Two different definitions are found in literature: The ability to communicate well with different people, including team members, managers, or clients, by providing the right information in a proper way, both oral and written (Deak, 2014b); the ability to convey information to be well understood and accepted (Ahmed, Capretz & Campbell, 2012). Both two definitions highlight the abilities to express oneself, rather than receiving information from others. Actually, communication is a two-way behaviour: passing your information to others while listening to others. Good communicators are always convincing, persuasive, good listeners and good presenters. Besides, English/foreign languages skills are a generic/auxiliary skill for daily communication. They were highlighted as the most important soft skills by Matturro (2013).

Communication skills are considered as one of the top S&Cs required of testers for several reasons, which is found to be cited in several different studies. Ahmed et al. (Ahmed et al., 2013) analysed the job adverts in four different portals and determined communication skills as the only skill in high demand. Other skills are found to be in medium or low demand. Seven over eleven test professionals interviewed by Deak (Deak, 2014b) highlighted the importance of communication skills in providing the right information to others in an appropriate way. Testers need to "communicate with different project team members, managers, sometimes even clients" and "report and discuss the bugs with developers" (Deak, 2014b). It may help handle conflicts in the team and make the development team more active (Whittaker, 2000). A tester was considered as the person associated with unwelcome "bugs". When testers discuss and report defects to different people, conflicts may arise easily if information is not well conveyed (Rivera-Ibarra et al., 2010). So, better communication skills can make testers easier to be understood and accepted (Deak, 2014b). It identified "efficient communication" as the most cited characteristics of a high performance team.

Although in most literature, generic communication skills are mentioned, there are some sub-skills. Good oral/written English (or Language) is highlighted as the No.1 skill in job adverts(Matturro, 2013). Although the importance is not explicitly explained, the importance of language in daily communication is evident. Language is one of the most effective ways for communication, helping to convey and to receive information. The questionnaire conducted by Holtkamp et al. (Holtkamp et al., 2015) shows another two important communication skills, namely "Ability to listen to others and consider their thoughts" and "Ability to communicate sensitively, taking into account other personalities and cultures". Those are also carefully considered in the proposed skill hierarchy.

Teamwork

Teamwork has two similar items: team player and collaborative skills. It is defined as the ability to work effectively in a team context and contribute positively to the team goal (Ahmed, Capretz & Campbell, 2012). One key responsibility of testers is to help the developer team deliver high-quality products (Cerqueira, 2016). They have to work cooperatively with different stakeholders, understanding and reviewing various artefacts (Rivera-Ibarra et al., 2010). Collaboration is important in improving team performance and productivity during the development process, especially in distributed software development. Good team players are expected to improve team performance and achieve team goals, rather than finishing tasks individually.

Experience in Testing Tools

Testing tools refer to testing products that can assist testers executing testing tasks automatically or semi-automatically (Lee et al., 2012). Operating the tools of testing is a common method for testers to assure the product quality (Lee et al., 2012). The survey (Lee et al., 2012) designed separate questionnaires for 22 companies and 20 experts. The results denote that tools are most frequently used in defect management and test reporting, followed by system test and integration test. But in the investigations conducted by Rivera-Ibarra (2010), the importance of this skill is reflected in dynamic analysis, coverage analysis and test case design.

Experience in Testing Techniques

Testing Techniques are needed in almost all test activities, especially when doing the experience-based and exploratory test (Itkonen et al., 2009). Beer (2008) declared that experience in testing techniques could help design test cases and select regression test cases based on the data collected from the testing activities of three companies, which is strengthened by Deak (2014b). Also, this skill can help when testers are involved in domains with volatile requirements (Deak, 2014b). They may know what to do under particular situations based on rich experience in testing.

Experience in Automation Test

It demonstrates the ability to write automated test scripts or apply specific tools to automate the test process.

The benefit from automation test is obvious. First, this relieves testers from executing the test cases manually and continuously. Puleio (2006) declared that automation keeps testing synchronised with development. Shaye showed the effectiveness of automation on continuous integration testing through practical experiments on an agile team(Shaye, 2008), which is strengthened by (Stolberg, 2009).Through the existing findings on software testing practices (Runeson, Andersson & Höst, 2003; Collins & de Lucena, 2012; Deak & Stalhane, 2013), it is an aspect that many organisations desired to improve. As automation testers are in high demand, it is advantage if a tester can do automation testing. Experience in automation testing can help testers build automation frameworks fast on a new project (Beer & Ramler, 2008).

Analytical, Problem-solving Skills

Analytical and Problem-solving Skills are usually talked together in investigations (Ahmed et al., 2013; Matturro, 2013). However, they belong to different cognitive skills. A good analyst can break the materials or concepts into parts and found the relevant logic or relationship between them (Anderson & Bloom, 2001). It focuses more on how to discover, differentiate or organise something through scientific methods. Problem solving skills can be defined as the ability to find solutions for a problem (John, 2009), or the ability to understand, analyse, and solve complex problems and make sensible decisions based on available information (Ahmed, Capretz & Campbell, 2012).

Ahmed et al. (2013) conducted an investigation on 650 job adverts (with 165 on software testing) and found the skill is associated with multiple testing tasks, including test plan, test case design and identification of defects.

Patience

Patience denotes the ability to endure something under challenging circumstances[7]. Several scenarios need testers' patience. Through the finding of interviews (Deak, 2014b), testers may encounter re-scheduled test plan or due day for the release, and have to work after regular office time; they may also find the version they just spent much time on testing is incorrect or outdated version; sometimes, they may face aggression or opposition because they keep on reporting different defects.

2.4.5 Research on Software Testing Courses

The importance of designing a testing course

One of the potential outputs is to create a blueprint for a testing course, developing the valuable and practical skills for future graduates.

In this thesis, the most important S&Cs for testers are investigated to benefit different roles, including educators and potential testers which may come from the students. For instance, it is necessary for educators to comprehend the requirements of employers (Petrova & Medlin, 2008) and the gaps between what employers want and what graduates have. They should figure out what skills needed are teachable and how to cultivate students related skills. This will enable students to sharp their skills and become welcomed by the industry.

The shortage of specialised professionals in testing can be associated with a lack

[7] https://en.wikipedia.org/wiki/Patience

of emphasis on education in ST: there are not enough courses nor enough effort to attract students with sufficient motivation, knowledge and practice (Valle et al., 2015). Through the survey (Garousi & Zhi, 2013), 50% respondents had received training on testing more than 20 hours per year, which is still far from enough. Also, these industry-based courses are an expensive way to learn all the testing knowledge/lessons after graduation and a testing course offered by tertiary institutions, in partnership with industry, can provide students with the simulation of actual working scenarios at an affordable cost (Shanwen & Zhengguang, 2012).

A good software testing course can narrow the gap between real requirements of the software industry and graduates' readiness (Almi, Rahman, Purusothaman & Sulaiman, 2011). With the fast-changing field of software testing, there is a critical shortage of qualified testers which may result from lack of related good education mechanism. It is becoming more and more indispensable to train and cultivate skilled testers.

Skills and Capabilities Emphasised in Testing Courses

Most existing courses would like to teach students techniques of developing software rather than breaking it (Bajaj & Balram, 2009; Meneely & Lucidi, 2013; Joshi & Desai, 2016). For example, Astigarraga et al. emphasised problem solving, debugging and analysis skills for the testing course (Astigarraga et al., 2010).

Early courses in ST focus on expositive teaching and related exercises, which shows the limitation in improving students' testing skills (Kaner & Padmanabhan, 2007). In 2008, Question-Driven Teaching Method (QDTM) in ST was proposed and had an initial effect in developing testers' skills. Problem based learning is an old practical teaching approach. It encourages students to work in groups and solve practical problems initiative. This can be used in teaching knowledge on software testing where problem-solving skills are of great importance. However, it is a challenge for teachers because they have to be actively involved in the learning process (Ikonen, Piironen, Saurén & Lankinen, 2009). This method becomes popular recently (Yinnan &

Xiaochi, 2011; Shanwen & Zhengguang, 2012; Bin & Shiming, 2013).

In the research (Bin & Shiming, 2013), the importance of practical exercise is elaborated and it has shown effectiveness in cultivating technical skills. Existing practical training can be divided into three categories: cooperation with enterprise (Shanwen & Zhengguang, 2012), using open-source projects (Fu, 2012), and exercise from other courses or papers (Joshi & Desai, 2016). Project-based pedagogy (Pradarelli, Nouet & Latorre, 2016) is one of them, which has shown its effectiveness in study of software engineering.

CDIO model is one of the most popular frameworks in modern software engineering education, including software testing (Yinnan & Xiaochi, 2011; Jia & Yang, 2013; Bin & Shiming, 2014). CDIO denotes Conceive, Develop, Implement and Operate (Yinnan & Xiaochi, 2011). This method consists of fundamental theory learning and practice in expertise of software engineering (Yinnan & Xiaochi, 2011). Seventeen different skills or abilities, not only technical skills, but also soft-learning skills, are covered in this education framework. Personal abilities, such as problem solving skills, teamwork and self-learning skill, could be trained by practice some projects in groups. Meanwhile, it can cultivate critical thinking by doubting the system. Students can also learn fast by sharing their understanding among classmates during the process. Many studies and experiments have indicated the effectiveness of this model in teaching software testing (Yinnan & Xiaochi, 2011; Jia & Yang, 2013; Bin & Shiming, 2014). It can systematically improve the technical skills, social skills and personal qualities of engineering students, helping them prepare for the challenges in real engineering jobs (Ikonen et al., 2009).

Besides, as many agile teams value individuals and interactions, some experiments were designed to cultivate testers those skills, such as English proficiency, oral and written communication skills (Valentin, Carvalho & Barreto, 2015)

To guide the design of a course, it is helpful to first understand any NZ-specific findings through the investigation conducted on current practices of NZ. There are several up-to-date and useful nationwide surveys on current practices of software testing available (Garousi & Zhi, 2013; Garousi et al., 2015; Garousi & Mäntylä, 2016), which can provide us with some inspiration to design our interviews.

2.4.6 Skill levels

Skills have different levels, which reflect the varying degrees of mastering the skills. Standards of different skills are important, and they may be the most obvious difference among different seniority for testers or other roles.

Many studies rank the importance of S&Cs in quantitative order based on frequencies (Ahmed et al., 2013; Deak, 2014b; Holtkamp et al., 2015). Based on the percentage of job counts that contains specific skill(Ahmed et al., 2013), the skills extracted from online job adverts were classified into three levels: High Demand (> 66%), Moderate Demand (> 33% and ≤ 66%) and Low Demand (≤ 33%), based on the percentage of job counts that contains specific skill. Holtkamp et al. (2015) conducted a quantitative approach: Delphi study, ranking the competencies into 1-7 (1 = not at all important and 7 = extremely important).

For cognitive skills, the well-known taxonomy is Bloom's Taxonomy (Anderson & Bloom, 2001), which was revised by Anderson and Krathwohl. It is originally used for permeated teaching and instructional planning (Krathwohl, 2016).

In Bloom's Taxonomy, knowledge in different levels can be associated with cognitive levels. Knowledge has four different types and levels: factual (facts, terminology, details, etc.), conceptual (theories, models, principles, etc.), procedural (methodologies, techniques, algorithms, etc.) and meta-cognitive (strategic or reflective knowledge) (Krathwohl, 2016). Meanwhile, cognitive skills are divided into six levels, namely remembering, understanding, applying, analysing, evaluating and creating. Common cognitive skills found for testers are analytical, problem-solving skills, which map the levels analysing and creating respectively.

The descriptions of cognitive levels can be found in Table B.3 in Appendix B.

2.4.7 Career Path of Software Tester

Only one article on the career path of testers is found. It defined five different roles in ST, hoping to support testers "benchmark" themselves (Weyuker et al., 2000).

Five role categories in the different level were simply described as follows:

- *Software testers*: an apprentice role. They are those who have just graduated or have little testing experience (usually less than two years).
- *Software test engineers*: mastery of core testing skills and knowledge, both engineering-related and testing-specific skills.
- *Software test specialist*: expert in one or more particular area. They may have several years of technical testing experience and are more professional than test engineers. Ability in judgement and evaluation are required to make testing decisions. Skills in test tools and automation, as well as those in test environment architecture, are needed.
- *Lead test specialist*: expert with some leadership ability in one or more technical testing area. To be a Lead specialist, one can either in-depth knowledge in a specific area, such as test automation, performance testing and security testing, broaden their expertise to a different area. Besides, they should have excellent leading skill in various testing activities.
- *Enterprise software test engineer*: expert on different technologies and testing practices. They are always responsible for strategic decisions.

This is a simple framework for testers and some of the terminologies are not so commonly used anymore. Besides, there is no evidence or data support the definitions.

2.5 Proposed Research

During this literature review, it has been found that more studies focus on software development, rather than software testing (Ryan & O'connor, 2009; Rivera-Ibarra et al., 2010; Matturro, 2013). Table B.1 shows demographic information of the studies obtained from the literature: In the existing studies which contain the investigation on skills of testers, only half of them focus on the skills of testing relevant roles (Beer & Ramler, 2008; Shah et al., 2014; Deak et al., 2016). Others are studies on the skills of different roles in development team or software engineering where testers are mentioned only in a small part of the whole study.

The research questions are inspired from the literature review. According to the recognition of the main researcher and supervisors, different countries may have different trend of software practices and context of industry. The investigation in this thesis will only be conducted in NZ to reduce the difference caused by the big context.

2.6 Conclusion

This chapter presents the approaches for searching existing information on this topic and findings obtained from the literature review.

Software testing has changed a lot in regards to the techniques and practices. Software testers have changed from the "filters" and "destroyer" of software, but "guard" of the products. Their responsibilities are not limited to discover bugs but also prevent bugs from sources.

The expected S&Cs for testers increase due to their responsibilities and associated way of working in the team. The increase of skill and capabilities requirements has caused the shortage of highly-skilled testers. Employers cannot recruit enough qualified testers and testers lack guidance on skill-set needed for the current industry. So investigations are designed to bridge the gap.

Definitions and explanations of the key concepts, including testers, S&Cs, are elaborated to clarify the subjects and scope of the research.

During the process of LR, no specific frameworks of S&Cs for testers are found, which will be one of the outputs of this research. Through LR, several highly ranked skills are identified, including domain knowledge, communication skills, teamwork, experience in testing tools etc.. The associated tasks or responsibilities are extracted from different studies and presented.

Some of these skills may be sharped through training and education. One of the most common skill model used in the current testing course is the CDIO model which includes 17 different skills and abilities. This can be used to guide the design of a practical testing course based on the finding of this research.

The proposed research will not only strengthen the current findings on the S&Cs for any testers, but also for different roles and seniority. Data about their importance will be collected to support the findings on the topic.

The findings from LR are used to form the research questions and compare the finding from the following investigations.

Chapter 3

Research Design and Implementation

3.1 Research Overview

The main aim of this research is to identify the expected generic S&Cs for testers in software testing in NZ, as well as gain a shared understanding of the S&Cs specific to different testing roles and seniority levels.

These aims will be achieved by answering the following research questions:

RQ1: What are the most important generic skills and capabilities needed by any tester, and why?

RQ2: What are the most important skills and capabilities that are specific for different testing roles, and why?

RQ3: What are the most important skills and capabilities of testers in different levels, and why?

I have taken a two-pronged approach to answering these questions. Firstly, online job adverts for testers will be used as a source of data since they contain information on capabilities and skills for various testing roles and seniority levels. Extracting this information from the job adverts will provide an overall list of S&Cs relevant to testers. These data can then be grouped and categorised to offer a structured model of relevant

S&Cs. This list will also be grouped according to tester roles and seniority levels to answer RQ2 and RQ3. The categories of roles and seniority levels to use for grouping will be extracted from the job adverts also. RQ1 will be answered by identifying the S&Cs that are common to *all* roles and seniority levels. The ranking of S&Cs by *importance* mentioned in all research questions will be based on the frequency of each skill and capability from the sample of job adverts used. *Why* specific testing S&Cs are important (or not) will also be extracted from the job adverts where possible. This is discussed in more detail in section 3.3. The list of S&Cs from the analysis of the job adverts will also be used to structure some of the interview questions, as explained briefly in the next section.

Using job adverts as the data gives the S&Cs from the perspective of the advert writers, who are likely to be a Test Manager, someone from Human Resources (HR) department or someone from an employment agency. The second part of the research seeks answers to the research questions from the perspective of test practitioners. The test practitioners are interviewed to elicit their view of the importance of the different S&Cs for different tester roles and seniority levels. The data from the interviews will be gathered using both semi-structured questions as well as structured questions based on Likert scales. Interviewees will be asked to indicate their view on the importance of each of the S&Cs identified from the job advert using Likert scales. The semi-structured questions will relate to the interviewee's experience as a tester and will be analysed using thematic analysis to identify patterns. There are two major expected outcomes from the interviews.

1. It can be used to triangulate the findings from two different perspectives of job advert writers and test practitioners. If the results agree, then this adds weight to how convincing they are. if there is some misalignment in the findings then this highlights areas that may need some attention for job advert writers.

2. Interviews provide an opportunity to understand *why* specific skills are considered more, or less, important. This information will be gathered using a think-out-loud protocol with the interviewees while they completed the structured questions in the interview. Some insights about the reasons for the importance of some skills for different roles and seniority levels were also expected from the semi-structured part of the interviews.

3. The background of the interviewees and their organisations will also be collected to provide a context for their answers.

In summary, two data sources are chosen in this thesis: a sample of job adverts and interviews. Job adverts will be collected to generate a conceptual model of testers' S&Cs, as well as their roles and seniority levels, and discover some patterns for the topic. Based on the initial findings from job adverts, interview questions will be generated and analysed to triangulate those findings, as well as get an in-depth understanding on the reasons for the importance of S&Cs. Figure 3.1 shows the entire process of the research.

To conduct the research, a plan is made to guide the activities in the research process. One of the most important steps is to choose research methods. The selection of research methods for data collection and analysis should help to achieve the research aims and objectives, and lead to answering any research questions.

As job adverts are textual data, particular analysis methods are needed to extract and code the required S&Cs needed for testers. The importance of each S&C is hard to obtain directly from the description of specific content because many job adverts in the sample did not have indications for the importance of each S&C. However, the importance can be identified by counting the occurrence frequency of its codes, which is a common use of Content Analysis (CA) (Hammond & Wellington, 2013; Kondracki, Wellman & Amundson, 2002). Two types of CA will be performed during the process

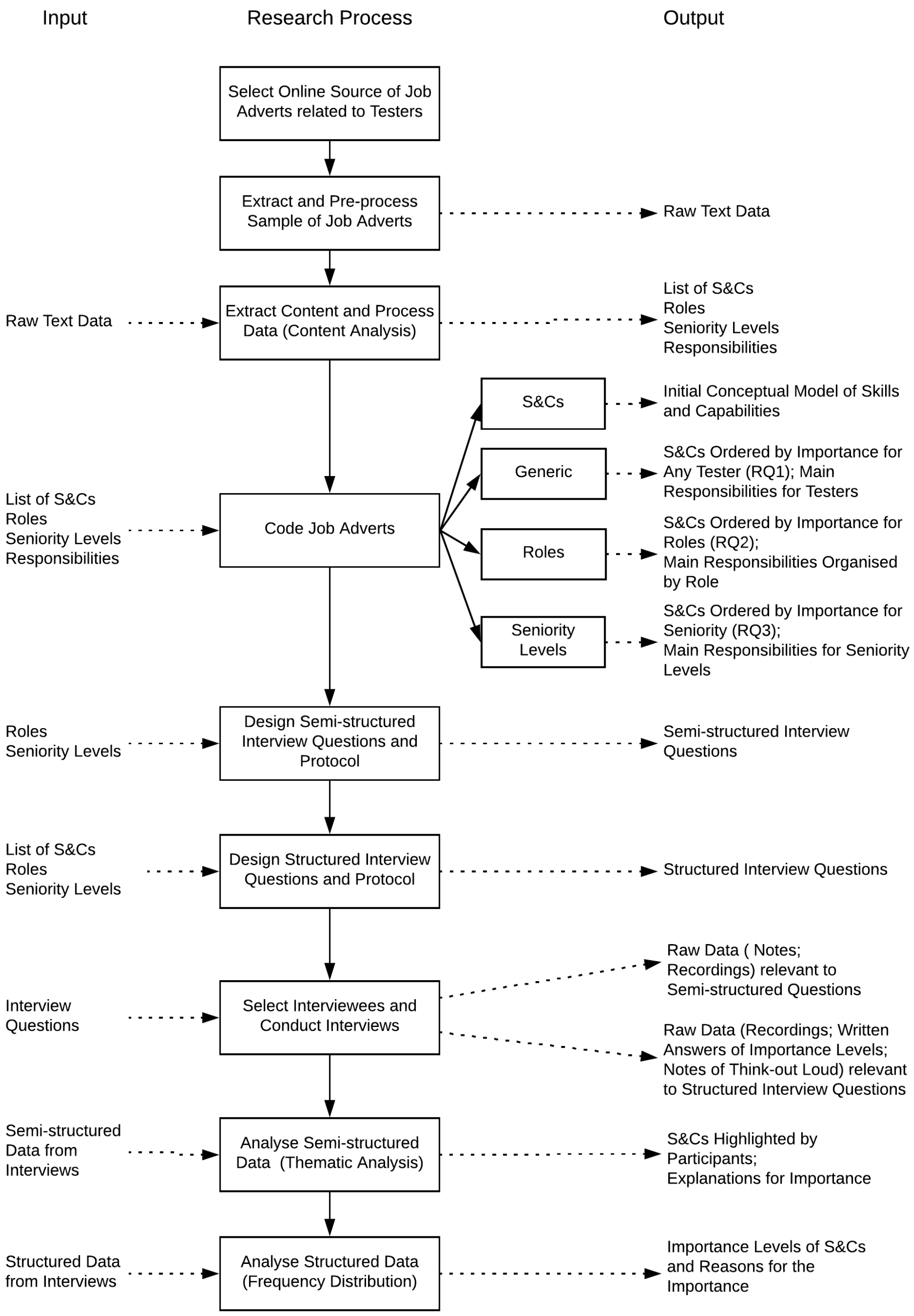

Input
Research Process
Output
Select Online Source of Job Adverts related to Testers
Extract and Pre-process Sample of Job Adverts
Raw Text Data
Raw Text Data
Extract Content and Process Data (Content Analysis)
List of S&Cs
Roles
Seniority Levels
Responsibilities
S&Cs
Initial Conceptual Model of Skills and Capabilities
Generic
S&Cs Ordered by Importance for Any Tester (RQ1); Main Responsibilities for Testers
List of S&Cs
Roles
Seniority Levels
Responsibilities
Code Job Adverts
Roles
S&Cs Ordered by Importance for Roles (RQ2);
Main Responsibilities Organised by Role
Seniority Levels
S&Cs Ordered by Importance for Seniority (RQ3);
Main Responsibilities for Seniority Levels
Roles
Seniority Levels
Design Semi-structured Interview Questions and Protocol
Semi-structured Interview Questions
List of S&Cs
Roles
Seniority Levels
Design Structured Interview Questions and Protocol
Structured Interview Questions
Raw Data (Notes; Recordings) relevant to Semi-structured Questions
Interview Questions
Select Interviewees and Conduct Interviews
Raw Data (Recordings; Written Answers of Importance Levels; Notes of Think-out Loud) relevant to Structured Interview Questions
Semi-structured Data from Interviews
Analyse Semi-structured Data (Thematic Analysis)
S&Cs Highlighted by Participants;
Explanations for Importance
Structured Data from Interviews
Analyse Structured Data (Frequency Distribution)
Importance Levels of S&Cs and Reasons for the Importance

Figure 3.1: The Process of Research

of coding the S&Cs in job adverts. At first, Directed Content Analysis (DCA) is used to extract role, seniority levels, as well as content blocks describing S&Cs and responsibilities in the sample of job adverts. DCA is employed because there are many indicative expressions in the job adverts, which can be used for extracting the useful text for particular keywords (role, seniority levels, S&Cs and responsibilities). This will be described in more detail in the following section. Then, Summative Content Analysis (SCA) has applied to code the synonyms and categorise them. SCA can analyse context by not only counting word frequencies (Kondracki et al., 2002), but also latent content analysis (Hsieh & Shannon, 2005). As job adverts are written by various recruiters and different words are used to express the skills and capabilities they want. Alternative expressions and synonyms should be identified before counting the frequency.

The interview questions consist of two main parts: semi-structured and structured. For semi-structured parts, thematic analysis is chosen because interview questions are designed and structured by research questions and data can be extracted and grouped by different themes. Besides, thematic Analysis is widely used for interview data (Vaismoradi, Turunen & Bondas, 2013; Deak, 2014b; Matturro, 2013; Shah et al., 2014). For the structured questions, a five-point Likert scale is used based on analysis of several studies (Garland, 1991; Clason & Dormody, 1994; Allen & Seaman, 2007). As mean and standard deviation is invalid for data from Likert scale, two norms are employed in this thesis. Frequency, or count occurring in response category, represents the distribution of importance levels, while median shows the common agreement on the importance of specific skills and capabilities.

3.2 Research using Job Adverts

Text data were chosen from online job adverts to identify different roles and levels of testers, as well as extract the most favourable S&Cs employers want. The frequency of occurrence was used to reflect the importance of extracted S&Cs. They were used to answer all the RQs from the employers' point of view.

3.2.1 Why Job Adverts As a Data Source

Job adverts were employed to understand the S&Cs needed for testers in a few studies (Ahmed, Capretz & Campbell, 2012; Matturro, 2013). The reasons for selecting job adverts to reflect the most favourable S&Cs employer want are discussed in this subsection.

Job adverts are well known as an indicator of short and mid-term market demands; they can reflect what the industry wants, or what employers believe they want (Clarke & Braun, 2014). They clearly express recruiters' viewpoints regarding the skills they need testers to have. No effective way was found to verify who is responsible for writing the job adverts. They may be written by a Human Resource (HR), test manager or even a consultant in job hunting. This thesis assumes that an employer is a person who is responsible for employing testers and writing the job adverts, and those adverts can reflect the expectation of employers. The findings from job adverts can be used to develop a conceptual framework of S&Cs, as well as guide the design of the interview questions. Job adverts have other advantages that there is a large volume of data freely and easily accessible online for extraction and further processing.

All data scraped from the job advert websites will be stored electronically as evidence of the research process and for possible future research.

A limitation of data collection from job adverts is that the data represents a snapshot of the job adverts over the period the data are collected.

3.2.2 Design of Job Adverts Analysis

The plan for the analysis of job adverts is elaborated in Fig 3.2.

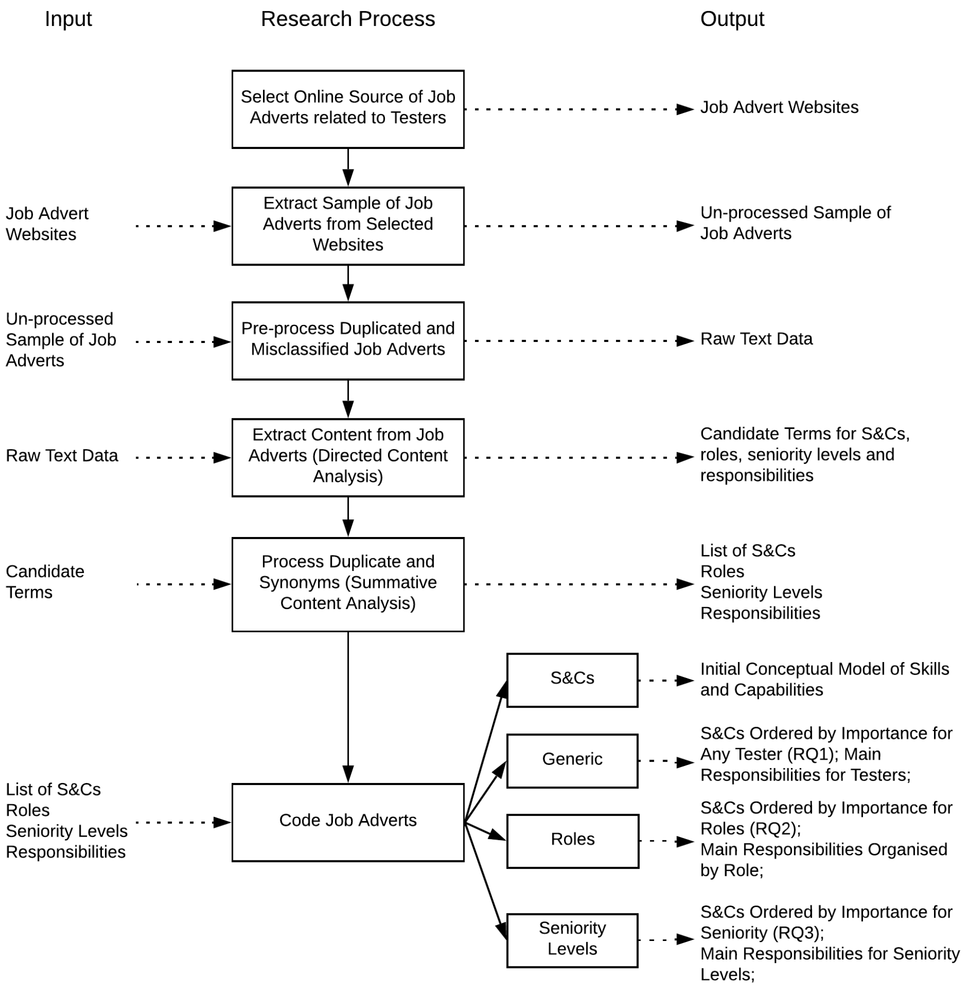

Input
Research Process
Output
Select Online Source of Job Adverts related to Testers
Job Advert Websites
Job Advert Websites
Extract Sample of Job Adverts from Selected Websites
Un-processed Sample of Job Adverts
Un-processed Sample of Job Adverts
Pre-process Duplicated and Misclassified Job Adverts
Raw Text Data
Raw Text Data
Extract Content from Job Adverts (Directed Content Analysis)
Candidate Terms for S&Cs, roles, seniority levels and responsibilities
Candidate Terms
Process Duplicate and Synonyms (Summative Content Analysis)
List of S&Cs
Roles
Seniority Levels
Responsibilities
S&Cs
Initial Conceptual Model of Skills and Capabilities
Generic
S&Cs Ordered by Importance for Any Tester (RQ1); Main Responsibilities for Testers;
List of S&Cs
Roles
Seniority Levels
Responsibilities
Code Job Adverts
Roles
S&Cs Ordered by Importance for Roles (RQ2); Main Responsibilities Organised by Role;
Seniority Levels
S&Cs Ordered by Importance for Seniority (RQ3); Main Responsibilities for Seniority Levels;

Figure 3.2: The Process of Job Analysis

It consists of several stages:

1. Data source selection: Search and select the websites that contain job adverts for testers.

2. Data extraction and pre-processing: Extract online job adverts for analysis and pre-process the invalid job adverts by eliminating duplicate and misclassified ones.

3. Content extraction and analysis: First, extract content blocks from raw data using the proposed content analysis methods. Then, process the codes and group them into different categories. Testing roles, seniority levels, responsibilities, as well as the classification of S&Cs, can be identified.

4. Code data to answer RQs: A conceptual model of S&Cs can be generated and the related frequencies are counted.

5. Analysis of findings: Interesting patterns are identified in the data.

3.2.3 Data Source Selection

It is quite important to identify and select the sources for job adverts. The investigation requires many samples to build the model and determine some pattern.

There are four steps for the source selection: set up criteria for websites, search candidate websites for job recruitment, select from candidates lists and determine sources.

Setting up criteria for websites

Two standards were set for candidate websites.

- Only NZ recruitment advertisements should be collected. The investigations conducted in this thesis are based on NZ, which limited the scope of sources.

- Websites which have massive job adverts in testing are preferred because the subjects of this research are testers.

Searching candidate websites

The main search string was "New Zealand job websites" and Google search engine was adopted. During the search process, a website[1] was found. This website introduces the main websites that post IT-related jobs continuously. So, the websites listed in

[1] https://www.careers.govt.nz/job-hunting/job-vacancy-and-recruitment-websites/

Table 3.1: Initial Statistics of Job Adverts in the Popular Job-related Websites in NZ

Jobs Websites	Total jobs in NZ	Information and Communication Technology	Software Testing & Quality Assurance
Seek	23946	2037	83
Trademe Jobs	21750	786	32
Hays	2917	88	9
myjobspace	4030	244	9
jobs.govt.nz	389	29	0

the website were searched one-by-one and the information about the job numbers are collected.

Selecting from candidates list and determining sources

Table 3.1 shows the total job numbers and those in testing. The data was collected on 11 March, 2017. As it was the weekend, no new job adverts appeared during the searching process. Five typical sites with more available jobs in NZ are listed in the Table 3.1: Seek[2], Trademe jobs[3], Hays[4], myjobspace[5] and jobs.govt.nz [6]. Other websites, including job monster, jobify and jobfeed, just offered a few jobs, and fewer jobs in IT. It is surprising that only two available test-related jobs were found in the IT-specific job site absoluteIT, which is claimed to be providing “best IT jobs in NZ”.

From Table 3.1, it is clear that Seek has the most job opportunities, as well as the most vacancies in software testing and quality assurance. Actually, Trademe jobs and Seek are the two most visited job sites in NZ, providing a wide range of vacancies. The total numbers of jobs offered were far more than other job sites. At the period of data collection, Seek listed many more opportunities in the area than Trademe Jobs, nearly triple at the time of data collecting. Besides, the number of jobs on Seek and Trademe available at different time periods remained quite steady over the following months.

[2] https://www.seek.co.nz
[3] http://www.trademe.co.nz/jobs
[4] https://www.hays.net.nz
[5] http://myjobspace.co.nz/
[6] https://jobs.govt.nz

So, Trademe jobs and Seek were chosen as the primary source of job adverts.

3.2.4 Data Extraction and Pre-processing

The process consists of two different stages in a circle: download job adverts and pre-process. The pre-process criteria include:

- Delete misclassified profiles by looking through the title and role introduction.
- Delete the duplicated jobs from both the title and body.
- Delete the incomplete job adverts with no specific skills and responsibilities. They are usually adverts with only an introduction or reference to other links.

The following is the detailed implementation of collecting the documents.

The main data were downloaded in Seek on 11 March. On Seek, the default classifications, rather than no particular keywords, were used to search. Here are the steps: Navigate to the main page of Seek NZ, click on the “Any Classification” drop down list in the main page of Seek, and the “Information & Communication Technology” group. A sub-category called “Testing & Quality Assurance” can be found, followed by the total number of this classification in the database. Next, if you choose "All New Zealand" in the “where” text box, and click "Search", the total job results and first 20 jobs are listed. Security is a particular aspect of testing. In Seek, security/penetration tester is classified into a category called "Security", rather than "Testing & Quality Assurance". So, it was not researched in this thesis.

Initially, 83 jobs were found. After downloading the jobs, two misclassified profiles were found. They both have the title of “Soil Tester”. As a result, only 81 complete job descriptions were retrieved with the help of NCapture, a subsidiary tool of NVIVO. After that, all the contents were copied into Word files for further processing and analysis.

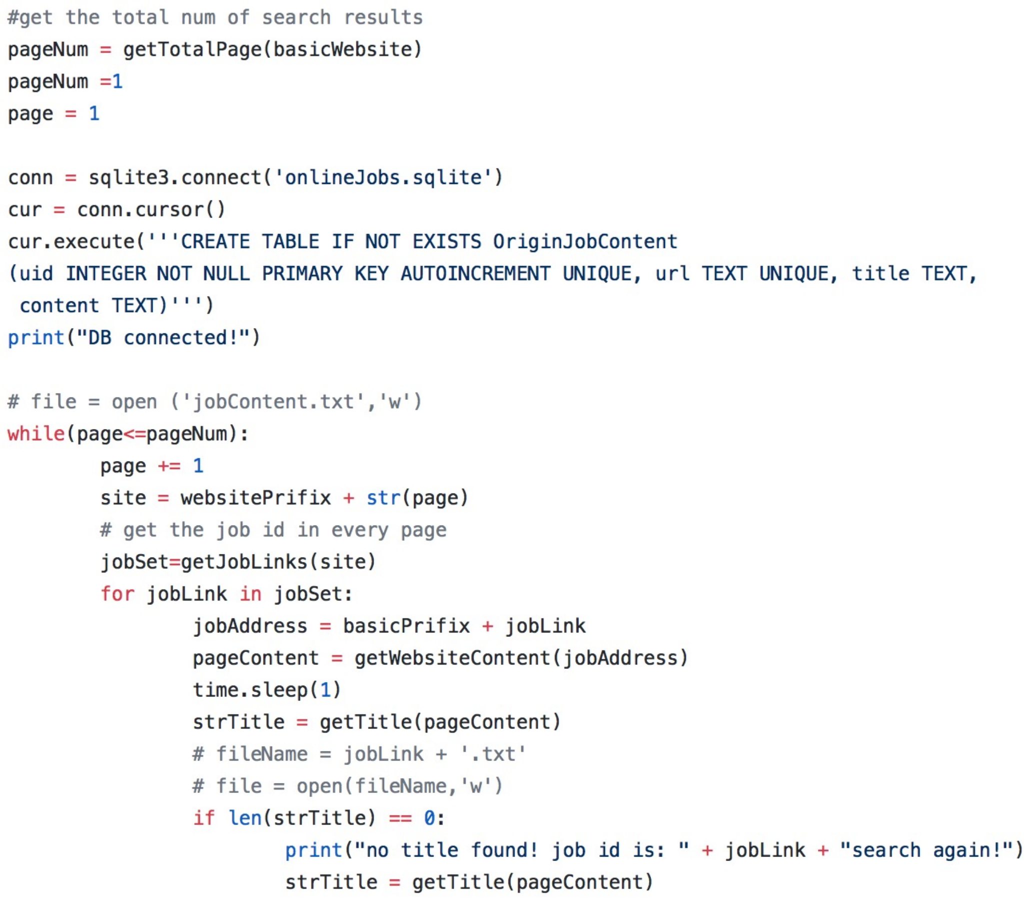
Some pre-processing steps were taken to clean the set of job adverts. First, the duplicated jobs were deleted manually. Two job adverts were considered as the same because they had the same company name, almost the same job role, as well as highly similar requirements and responsibilities. Six duplicated ones were eliminated. 75 job adverts from Seek were left. Then, after looking through all the reminders, one job advert with no specific skills and responsibilities listed was deleted because it has no valid information on S&Cs and was considered incomplete. Eventually, 74 job adverts were obtained on Seek.

A similar method was used to pre-process the jobs from Trademe. The job adverts in Trademe were collected on 13 Mar 2017. In Trademe Jobs, choose “IT” in the drop-down list of “All categories”, and “Testing” in the drop down list of “All subcategories”, click “Search Jobs”, all testing jobs were listed in the results. When searching through the whole of NZ, default values about regions and districts were used. Thirty two job descriptions were captured. After being compared with the contents of the job adverts, only four were found different from those in Seek and were added to the job list. Then, 78 job adverts were obtained for the following analysis.

During the early stage of this research, 46 job adverts were collected from Seek in 22 January, 2017. Keywords, such as "tester", "QA", and "testing", were used. This was a pilot study which helped the principal researcher to get familiar with the skills and responsibilities used in existing job adverts.

The items of skills and responsibilities mentioned in the jobs collected in January and March did not show many differences. So the jobs gathered in two different times were combined to increase the sample size. As most jobs posted on Seek have a lifespan of four to six weeks, the number of duplicated jobs were much less than that between Seek and Trademe Jobs in the same period. After pre-processing, 39 more job adverts were added to the original set. This means a total of 117 job samples were obtained.

Actually, the data can also be collected automatically by crawl techniques. An



```
#get the total num of search results
pageNum = getTotalPage(basicWebsite)
pageNum =1
page = 1

conn = sqlite3.connect('onlineJobs.sqlite')
cur = conn.cursor()
cur.execute('''CREATE TABLE IF NOT EXISTS OriginJobContent
(uid INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE, url TEXT UNIQUE, title TEXT,
 content TEXT)''')
print("DB connected!")

# file = open ('jobContent.txt','w')
while(page<=pageNum):
        page += 1
        site = websitePrifix + str(page)
        # get the job id in every page
        jobSet=getJobLinks(site)
        for jobLink in jobSet:
                jobAddress = basicPrifix + jobLink
                pageContent = getWebsiteContent(jobAddress)
                time.sleep(1)
                strTitle = getTitle(pageContent)
                # fileName = jobLink + '.txt'
                # file = open(fileName,'w')
                if len(strTitle) == 0:
                        print("no title found! job id is: " + jobLink + "search again!")
                        strTitle = getTitle(pageContent)
```

Figure 3.3: Snapshot of Source Code for Crawling Online Job Adverts

automated crawling script was implemented using Python 3.6. It can capture all job adverts in the categories described above and save the job information into files or a database. Figure 3.3 is a snapshot of the source code. The complete source code is available to download in Github [7].

3.2.5 Analysis Methods for Analysing Job Adverts

Job adverts are textual data. Qualitative methods were first used to extract the skills and related expressions before analysing the frequency of each skill and capability using qualitative analysis. Content Analysis (CA) was chosen because it is a systematic

[7] https://github.com/lynnliang/JobCrawler

method in coding and categorising textual information to explore trends and patterns of words, their frequencies and relationships, and so on (Descombe, 2003; Vaismoradi et al., 2013; Kanij et al., 2014). It usually focuses on interpreting the context by assessing intentions of specific content, which satisfies what we need to analysis job adverts. Similar to the studies (Hammond & Wellington, 2013; Kondracki et al., 2002), the words and synonyms of particular S&Cs were first extracted and classified before frequencies of them were counted.

General steps are described as follows:

- Read through the job samples and highlighted the text blocks.
- Code the highlighted textual information into a wide range of units, where each of these units has a labelled category. A new code is given when some text could not be categorised with the initial coding schema.
- Group units with equivalent expressions and refine the categories and code.

To be more detailed, two different content analysis approaches, namely DCA and SCA, were employed in this thesis. They are used to generate the categories of roles, seniority levels and S&Cs and analyse the most important of testers from the job adverts.

DCA, a conventional method of CA, can help to determine the initial coding schema using a deductive process. It was applied to extract content blocks that describe the S&Cs for various testing roles and seniority levels from the raw data, using the indicative words and expressions.

SCA focuses on analysing the content by counting word frequencies or latent content (Kondracki et al., 2002; Hsieh & Shannon, 2005). Alternative expressions of specific items were identified and grouped for data classification (Hsieh & Shannon, 2005) to provide a customised model of S&Cs for testers.

The detailed implementation of roles, seniority levels and S&Cs are elaborated as follows.

Analysis of roles and seniority levels

RQ2 and RQ3 ask what are the most important S&Cs for each testing role and seniority level. So, the most critical categories of role and seniority levels should be identified in order to code the S&Cs.

The roles and seniority levels were extracted together because they mainly exist in the job title or the introduction of the role in the body. This process can be described as follows:

1. Mark the content that contains the information of roles and seniorities
2. Extract the names and expressions that used for roles and seniority levels.
3. Group similar expressions, choose a common name and generate the initial categories.
4. Check the responsibilities and requirements of some samples with the fixed items, and adjust the classifications. This is a HC process of understanding and revising.

In this thesis, roles and seniority levels were extracted from the titles and induction part of job adverts at the same time. Job adverts usually include a role name for the person they want to recruit, which is usually expressed in the title. Sometimes, an exact role name is described in the introduction section.

As there were no official names or standards for testing roles, various role names or expressions were used for the same role in the job adverts. A common name was chosen as a label for each role. This makes the results clear and easy to analyse. SCA was employed to group roles of various job titles. Keywords and expressions related to role names were identified and coded manually. They were grouped based on the understanding of contextual information.

Table 3.2 summarises the tester role codes identified from the job adverts. Each code represents the category that a number of specific role labels found in the job adverts have been classified as. The classification is based on similarity of roles and role synonyms. The list of role labels that are assigned to different role codes are presented in Table C.2 in Appendix C. From the collected job samples, seven test roles are identified: Test Analyst (TA), Quality Assurance (QA), Automation Tester (AT), Performance Tester (PT), Integration Tester(InT), Product Test Analyst (PTA) and Firmware/HW Test Engineer (TE).

Some rules are applied to classify the roles:

- The description and requirements of the role. For example, many QA analysts, such as B18, B26, and B27, have quite similar responsibilities with Test Analyst.
- The different expressions used in the same job profile. For instance, "Automation Tester" is recognised as the same as "Automation Test Analyst" according to A04. For the "Automation Mobile Test Analyst" described in B05, they state the role as "A Senior Mobile Automation test specialist". All these were taken into account for the final classification.
- Common sense. First, recruiters were using different parts of speech for the same word root. Also, some words, like "software", were implicit in the job profile. So, "Automated Tester" and "Software Automation Tester" are regarded the same role as "Automation Tester".

However, the decision made on seniority levels was easier than test roles. The role levels have unified names; explicit words, including junior, intermediate, senior, lead and manager, are used in many job adverts. According to the job samples, lead and

Table 3.2: Different Roles Found in Job Adverts

Role Name	Names in job adverts
Automation Tester (AT)	Automation Tester; Automation Test Analyst; Automation Test Lead; Automation Test Engineer; Test Automation Engineer; Automation Engineer; Automation Mobile Test Analyst; Automation Specialist; Mobile Automation Tester; Selenium Test Automation Expert; Automated Test Manager; Development Engineer in Test; Technical Test Analyst;
Test Analyst (TA)	(Software) Test Analyst; Quality Assurance Analyst; Test Manager; (Software) Test Engineer; Software Tester; Automation & Functional Tester; QA Test Analyst; Test and Release Lead; Test Lead; Lead QA Engineer; Lead Tester; Quality Engineer; Software/ Devices Tester; Test Director; Embedded Test Engineer;
Product Test Analyst (PTA)	Dynamics CRM Test Analyst; SAP Test Analyst; Lead Product Tester; Test Analyst-Cards Switching; Web Applications Tester
Quality Assurance (QA)	National Lead for Quality; Quality Analyst; Quality Business Analyst; Software QA Engineer
Integration Tester (InT)	(Systems) Integration Test Lead; Systems Integration Tester;
Test Engineer (TE)	Firmware Test Engineer
Performance Tester (PA)	Performance and Volume Test Lead (P&V Test Lead); Performance Technical Test Analyst; Performance (Test) Engineer; Technical Tester expertise in load and performance testing

manager are explicitly included in the role name. For instance, "automation test lead", "performance test lead", "Lead Tester", etc., are used for test lead.

For those job adverts without explicit seniority levels, judgements are made based on the following considerations:

- Comparing the responsibilities and description of the role with the ones that have been confirmed. For example, juniors are usually asked to execute test cases, while seniors are expected to mentor juniors.

- Required experience in testing. For example, junior Testers refer to those who have little or no previous industry experience in testing, usually less than two years' experience in testing. Intermediate testers are likely to be those have two to five years' testing experience, while seniors more than four or five years.
- Those with two different levels in the job title are classified into both categories. Take B66 for example, the recruiter is expecting either an intermediate or senior. So it is counted for both intermediate and senior.
- Graduate testers are considered as junior for two reasons: both of them are new test practitioners with little or no testing experience; they are all supposed to be provided with sufficient training and learning opportunities.

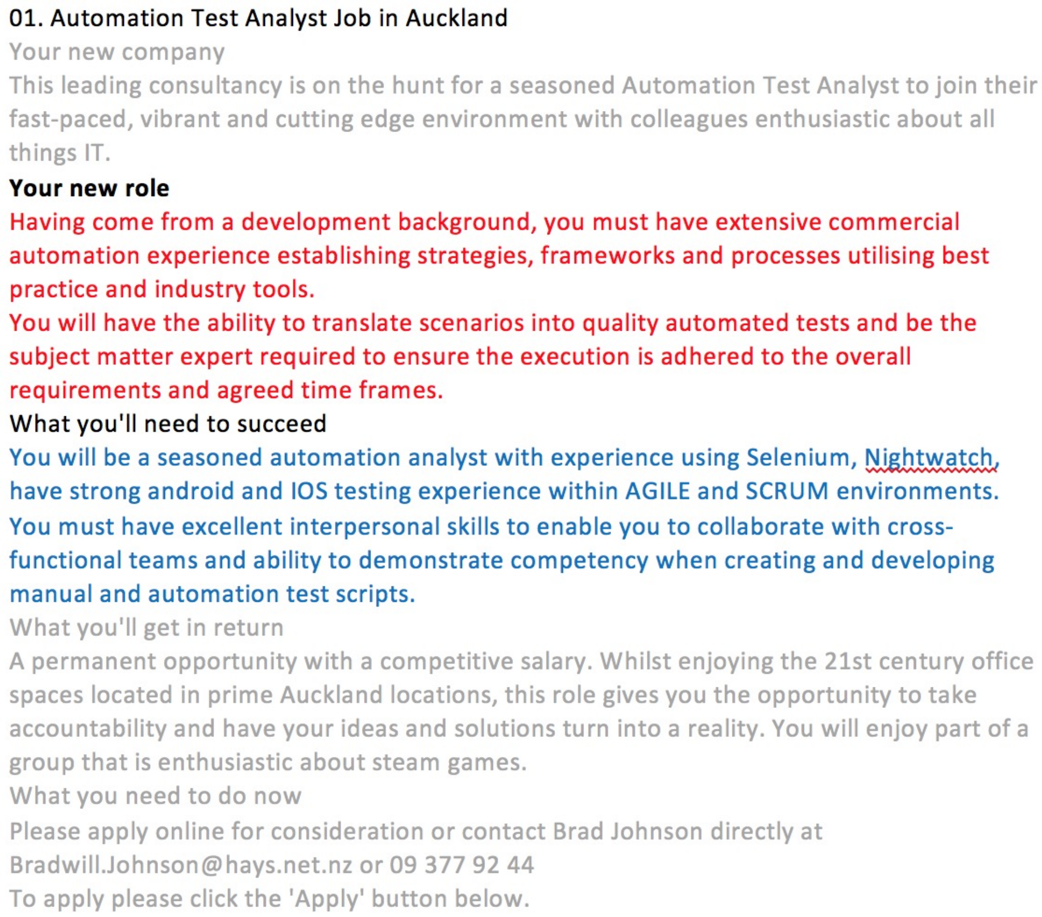
Analysis of skills and responsibilities

Select App for data analysis

As the jobs were captured using NCapture and imported into NVivo, the jobs were mainly ordered by the starting character of the job title. Then, they were copied into two word files. The data were planned to be analysed using NVivo, until the performance was found to not to be ideal. The operations in NVivo are not as flexible, and the free version for students was unable to build maps between different variables. Besides, it was too slow to run NVivo in the computer. So, Microsoft Excel was considered as the alternative app to NVivo.

Extract content blocks of skills and responsibilities from raw data

Most job adverts consist of several parts: the introduction of the company, main responsibilities, key skills and experience, the motivations of this job and contact information. Fig. 3.4 highlights the skills and responsibilities in an exemplary job profile. But still, some samples describe their requirements of skills, especially soft skills, in different ways. Some job adverts, such as B26, put some skills at the beginning



01. Automation Test Analyst Job in Auckland

Your new company

This leading consultancy is on the hunt for a seasoned Automation Test Analyst to join their fast-paced, vibrant and cutting edge environment with colleagues enthusiastic about all things IT.

Your new role

Having come from a development background, you must have extensive commercial automation experience establishing strategies, frameworks and processes utilising best practice and industry tools.

You will have the ability to translate scenarios into quality automated tests and be the subject matter expert required to ensure the execution is adhered to the overall requirements and agreed time frames.

What you'll need to succeed

You will be a seasoned automation analyst with experience using Selenium, Nightwatch, have strong android and IOS testing experience within AGILE and SCRUM environments.

You must have excellent interpersonal skills to enable you to collaborate with cross-functional teams and ability to demonstrate competency when creating and developing manual and automation test scripts.

What you'll get in return

A permanent opportunity with a competitive salary. Whilst enjoying the 21st century office spaces located in prime Auckland locations, this role gives you the opportunity to take accountability and have your ideas and solutions turn into a reality. You will enjoy part of a group that is enthusiastic about steam games.

What you need to do now

Please apply online for consideration or contact Brad Johnson directly at Bradwill.Johnson@hays.net.nz or 09 377 92 44

To apply please click the 'Apply' button below.

Figure 3.4: An example of job adverts

of the content to attract readers. Other job adverts, such as A05, stagger the skills and responsibilities.

During this process, DCA was applied to extract the content blocks for skills and responsibilities. Indicative words and expressions were identified for grouping.

Table 3.3 illustrated the key indicative words or phrases for skills and responsibilities presented in job adverts. Key terms, including “Skills & experience required”, “To be successful you will need”, “What we want from you?”, “To be successful in this role you will have”, “What are we looking for”, etc., are placed under the key skills. Occasionally, the skills are described in summary without explicit keywords. They were extracted manually to increase the accuracy of the statistics. Similarly, “Responsibilities

Table 3.3: Indicative Expressions for Skills and Responsibilities

Content	Key Indicated Expressions
Skills and Capabilities	key skills, skills needed, additional skills, here's a list of skills you'll need, key skills and experience, skills and experience required, technical skills and requirements, experience doing, preferred experience the selection criteria, skill/experience criteria essential, ideally you will have experience with * as much of the the right candidate, the ideal candidate will have, key requirements, role requirement, your background, your profile, about you, to be successful in this role, you are a * tester, you have, you care about, what you'll need what do I need, what you know, who you are to succeed, you will need to demonstrate, bonus you'll be able to
Responsibilities	your new role, the role responsibilities, require an experienced tester to, about your new role, in this role you will be, what will you do, about the role, what you'll be doing, duties & responsibilities, in this role you will…

include", "The Things you'll get to do", "position that will require the following to be considered", "What will you be doing?", "Your responsibilities will include", are indicators of responsibilities.

They are followed by steps of identifying the names of different items.

Organise data for analysis

The content of job samples is organised by an Excel file.

First, the whole source of job adverts is imported into the first worksheet called "OriginJobadverts". Then, all the source data are stored in a worksheet. Each can be identified by an ID. The IDs of jobs from January begin with an "A" while these from March are assigned with a "B". For example, the first job captured in January was marked as "A01", while those collected in March was labelled as "B01".

Then, the extracted content was copied into the corresponding column of "Expected skills" and "Responsibilities" in spreadsheets "JobData1" and "JobData2". Worksheet

"JobData1" contains data for those collected from January while "JobData2" comprises data from March. In these two spreadsheets, each row has a hyperlink pointing to the original source of job adverts, so that readers can easily track the source and validate the correctness. The two sheets are used to store the extract data for analysis, such as roles, seniority levels, content blocks about skills and responsibilities, extracted results for final counting.

Identify equivalent expressions

The preparation process is to find different skills and equivalent expressions for them.

In this thesis, the expressions that appeared in job adverts were grouped, and a standard and representative name among them was chosen as the skill name. The reasons for this are:

- The target of this research is to discover, rather than just using existing frameworks and skill names.
- Choosing those names in job adverts can keep the coherence of results. Skills may have various expressions. Similar skills, no matter find in LR or job adverts, can be used to generate search strings for each skill.
- A final comparison with all the findings from different investigations will show the gaps in the cognition of S&Cs.

To get a holistic understanding of the different expressions of specific skills, the main researcher read through the job content for a few times when collecting similar expressions for the same skills to obtain a general sense of the whole picture.

Initially, expressions of S&Cs were extracted from the raw data (see Table C.1 in Appendix C) and a total of 372 items were obtained. They were listed in alphabetical order first and grouped by a few rules afterwards. This is a complicated process of

SCA, to understand the contextual use of different skills. First, different expressions, such as different tense, voice and part of speech, were considered as the same item. Take "problem solving" as an example. it may be shown as "problem-solving, problem solving, or problem solver" etc. Second, detailed expressions were grouped gradually. For example, "SQL database admin skills", "SQL querying skills" and "SQL scripts" were grouped into "SQL" before them classified into "Database knowledge". There were also various tools and languages, they were all classified into different test activities. Items, such as "APM", "JMeter" and "LoadRunner" are for performance testing, while "selenium", "cucumber" and "SpecFlow" are for automation testing.

Then, equivalent expressions were added to avoid missing important items. For instance, "flexible" was identified as a synonym of "adaptable", so was "initiative" to "proactive". Most of the skill names were chosen from the job adverts, while some were summarised names made up by a series of sub-skills.

Table C.3 to C.5 in Appendix C shows the synonyms of each skill and capability in the proposed model. The first column is the unified name of S&Cs extracted from the job adverts. The second column contains a collection of the keywords and synonyms that are classified into the specific item. The third column denotes the frequencies found in the sample of adverts.

The identification of the S&Cs was a process of Hermeneutic Circle (HC) and the understanding about the S&Cs was revised in an iterative process (Fleming, Gaidys & Robb, n.d.). HC is a research approach associated closely with interpretative understanding (Boell & Cecez-Kecmanovic, 2010). The S&Cs were determined after several adjustments and discussions. They were classified into different groups based on the findings from LR, and the knowledge of the main researcher and supervisor. Then, an initial Framework of S&Cs was generated. The proposed framework was later used to conduct quantitative data analysis on the frequencies of those S&Cs. The results will be presented separately.

3.2.6 Generate data to Answer RQs

Office Visual Basic for Applications (VBA) [8] is an event-driven programming language which helps us to operate the Excel and other Office programs. It is widely used in our data analysis process, to avoid repeated manual operations. For example, regular expressions enable us to extract skills and responsibilities in the content, which were recorded in the cells next to the origin content.

The complete list of skills was divided into blocks with the high-level skill names and its sub-skills. Similar or equivalent expressions of that specific S&C found in literature review or job adverts were grouped.

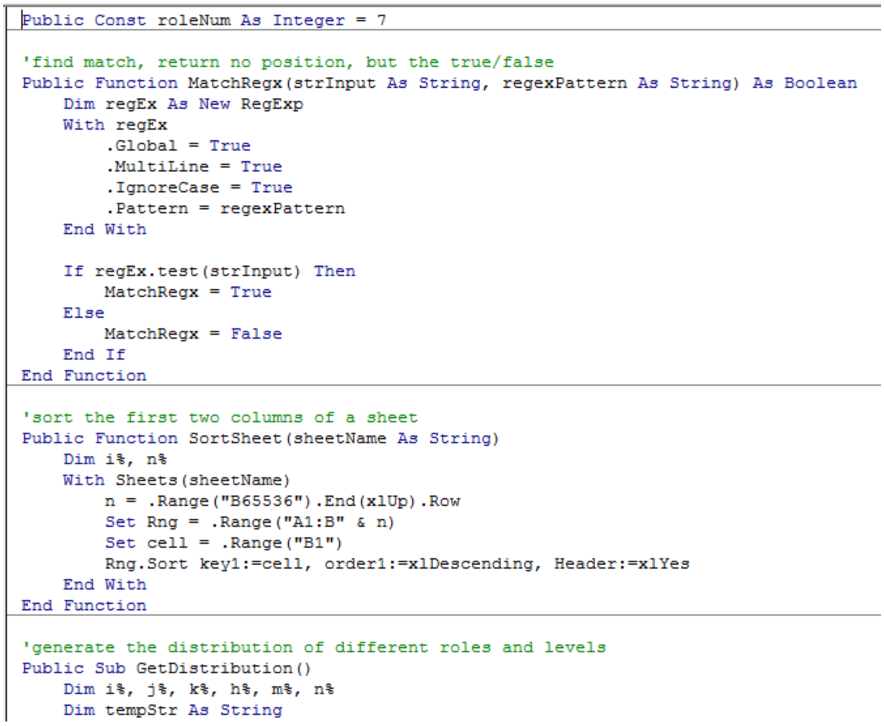
Then, frequencies of skills appearing in the job adverts were counted to find the most favourable S&Cs using VBA. The frequency stands for the number of job adverts which contain specific S&C. It is obtained using the following formula.

$$Frequency = \frac{Numberofjobadvertswiththeresponsibilityforthisrole}{Thetotalnumberofjobadvertsforthisrole}$$

In this thesis, multiple appearances of the same S&C in job adverts was counted as one. This can help to determine the relative importance of specific item compared with others appeared in the sample of adverts. Results were saved in different worksheets. Figure 3.5 is an example code using VBA. To click the button "Skill Frequency", all the skills included in each job profile were extracted and counted in different ways. To get the most value from the data, skills expectations were counted, both of high-level and low-level, for different roles and seniorities. Also, the frequency of how many companies claim themselves as agile and what is the particular skill-set of agile testers.

Similar operations were performed to responsibilities. The responsibilities for each of the roles and seniority levels were extracted by firstly coding each job advert with a role and seniority level and extracting the responsibilities from every job advert in

[8] https://msdn.microsoft.com/en-nz/vba/index



```
Public Const roleNum As Integer = 7

'find match, return no position, but the true/false
Public Function MatchRegx(strInput As String, regexPattern As String) As Boolean
    Dim regEx As New RegExp
    With regEx
        .Global = True
        .MultiLine = True
        .IgnoreCase = True
        .Pattern = regexPattern
    End With

    If regEx.test(strInput) Then
        MatchRegx = True
    Else
        MatchRegx = False
    End If
End Function

'sort the first two columns of a sheet
Public Function SortSheet(sheetName As String)
    Dim i%, n%
    With Sheets(sheetName)
        n = .Range("B65536").End(xlUp).Row
        Set Rng = .Range("A1:B" & n)
        Set cell = .Range("B1")
        Rng.Sort key1:=cell, order1:=xlDescending, Header:=xlYes
    End With
End Function

'generate the distribution of different roles and levels
Public Sub GetDistribution()
    Dim i%, j%, k%, h%, m%, n%
    Dim tempStr As String
```

Figure 3.5: Snapshot of Source Code using VBA

the sample. The job responsibilities from the job adverts were also categorised into responsibility codes (see Tables C.6 and C.7 in Appendix C). The ten responsibilities with the highest frequency were then used to assign to the job roles. There were a few job adverts that had TA responsibilities sought for an AT job and vice versa and these were removed by considering only the high-frequency responsibilities for each role.

As the final results were generated, analysis can be conducted to answer the RQs. This will be presented in the next Chapter.

3.3 Interview-Based Research

One of the aims to do the interviews is to get a deep understanding of what S&Cs work within the NZ industry context and the reasons for their importance. Although conclusions cannot be drawn just from a few interviews, a deep insight into the topic from the test professionals' viewpoint was obtained and patterns on this topic were found.

3.3.1 Why Interview As a Data Source

Interviews were used as a source for several reasons:

- They can provide data to answer the RQs from the perspective of test practitioners.
- They can help explain the importance levels by well-designed questions.
- Similar studies used interviews to collect data (Deak, 2014b; Matturro et al., 2015; Deak et al., 2016)

Interviews can be defined as conversations between the researcher and those being researched, namely participants, subjects or "interviewees" (Hammond & Wellington, 2013). These conversations are usually taken under the governance of ethical rules (Hammond & Wellington, 2013).

There are three main types: structured, unstructured and semi-structured (Runeson & Höst, 2009).

The main part of the interview questions is semi-structured. Semi-structured interviews are conversations in which you have fixed topics and a set of questions to find out, while the conversations vary between participants (Fylan, 2005). They were widely used in software engineering research (Easterbrook, Singer, Storey & Damian, 2008).

Semi-structured interviews have several advantages. They are both manageable and flexible (Hammond & Wellington, 2013). They can make the interviews both descriptive and explanatory, which is the main reason for choosing this interview type. In a semi-structured interview, questions cannot be asked in fixed order, decided while the conversation is progressing. The interviewer can probe maximum information from one question while focusing on the target of the study topic. Also, it contributes positively to the trust building between the interviewer and interviewee (Buchan, Ekadharmawan & MacDonell, 2009). Participants are more likely to feel easy and comfortable discussing their experience and reflections on the research topic. Besides, the data collected in this way will be easy to analyse, because the useful information are grouped by different themes and the answers can be extracted easily using existing approaches.

On the other hand, structured questions were designed to collect interviewees' view on the important levels of S&Cs in the proposed model, which might not be extracted in job adverts. Participants were asked to think out loud to help understand why each S&C was considered more or less important when they are filling in the importance levels of Likert scale. The structured questions can ensure each S&C in the proposed levels has an assigned value of Likert scale from the perspective of test professionals.

In summary, mixed interviews including both semi-structured and structured interview questions were designed. Semi-structured interview questions were designed to explore the most valuable S&Cs of testers that were not included in job adverts and collect the context of interviewees and the companies they were working. It enables the primary researcher to focus on the interview questions, as well as allowing interviewees to express their perceptions and real thoughts. Besides, structured questions were designed to collect interviewees' view on the importance levels of the S&Cs in the proposed conceptual model and guide them to speak out the reasons why each S&C is more or less important.

Due to the limited size of samples and the nature of the importance levels, the data were analysed mainly using qualitative methodologies.

3.3.2 Design of Interviews Analysis

The plan of interviews analysis is presented in Figure 3.6. First, interview questions are designed according to the RQs. Then, before applying for ethics approval, documents,

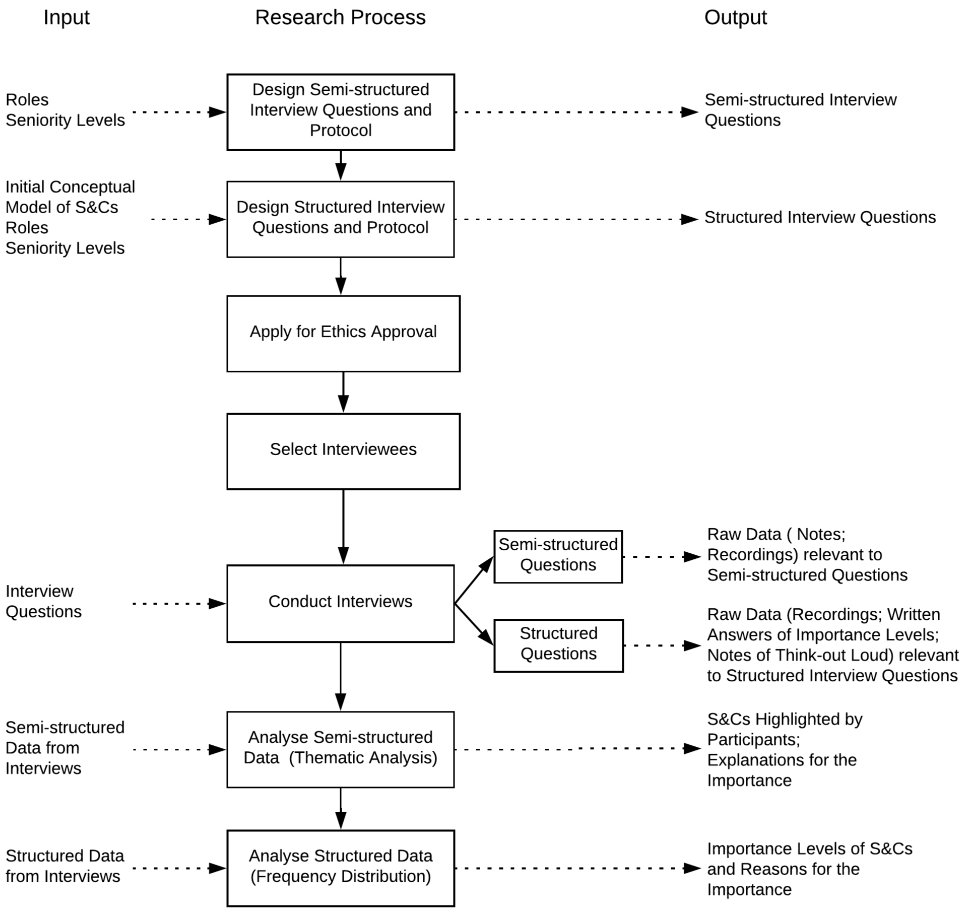

Input
Research Process
Output
Roles
Seniority Levels
Design Semi-structured Interview Questions and Protocol
Semi-structured Interview Questions
Initial Conceptual Model of S&Cs
Roles
Seniority Levels
Design Structured Interview Questions and Protocol
Structured Interview Questions
Apply for Ethics Approval
Select Interviewees
Interview Questions
Conduct Interviews
Semi-structured Questions
Raw Data (Notes; Recordings) relevant to Semi-structured Questions
Structured Questions
Raw Data (Recordings; Written Answers of Importance Levels; Notes of Think-out Loud) relevant to Structured Interview Questions
Semi-structured Data from Interviews
Analyse Semi-structured Data (Thematic Analysis)
S&Cs Highlighted by Participants;
Explanations for the Importance
Structured Data from Interviews
Analyse Structured Data (Frequency Distribution)
Importance Levels of S&Cs and Reasons for the Importance

Figure 3.6: The Design of Interviews Analysis

including the Participant Information Sheet and Consent Form, are prepared. This is followed by conducting the interviews. To save time, notes will be taken during and after the interviews. Data can be grouped into different themes and analysed using qualitative data analysis. Finally, the finding is used to triangulate the findings from the perspective of employers and test practitioners and understand the reasons why the S&Cs are more or less important.

Table 3.4: Mapping between Research Questions and Interview Questions

Research Questions (RQs)	Interview Questions
Background	Q1-11
RQ1	Q13, Q17
RQ2	Q12, Q17
RQ3	Q14-16, Q17

3.3.3 Design of Interview Questions

Findings from literature review and job advertisements analysis were used to form the initial interviews questions and group them. The interview questions were determined after a number of discussions with the supervisor and revised after a pilot interview on one test professional. The pilot interview was informal but provided us with some insight into the amount of time which might be spent on each part, as well as highlighting the need to refine some of the structured questions.

A data collection form was used to structure the interview answers and ensure all questions were answered at some stages in the interview.

The interview questions were mapped to the research questions. The whole question set consists of several parts (see Table 3.4). In this thesis, the interview questions begin with "Q", and the research questions begin with "RQ". Detailed interview questions can be found in Section D.1 in Appendix D.

Part one of the semi-structured interview is about the background of the participants and their organisations. It consists of the organisational and team context (Q1, Q4-6), software development process (Q2-3) and the background and experience of the participants (Q7-11). These questions are useful for the reader to understand the contexts of the interviewees.

Subsequent questions are about the opinions and views of the interviewees and so during the process, the main researchers are not expected to express their own opinions to influence the conversation and results.

Table 3.5: Mapping between Likert Scale and Indicated Importance

Five-point Likert Scale	Indicated importance
5	Very Important
4	Important
3	Moderately Important
2	Slightly Important
1	Not Important

Part two answers RQ1. It asked interviewees about the generic S&Cs that are most important for testers, independent of role and seniority level.

Part three relates to RQ2 and was designed to collect the interviewee's opinions of what makes the interviewee a good tester in their current role. It should reflect the most important S&Cs related to the current role of the participant.

RQ3 is addressed in Part four of the interview questions. It is about the differences in S&Cs at different seniority levels. This part can give us insights into the various skill requirements for different seniority levels and what skills should be enhanced to shift from one level to another.

The next part of the interview involves answering structured questions about the levels of importance of different S&Cs for their role. These questions also served as a trigger for the interviewees to speak out loud their reasons for selecting a particular importance level for each S&C.

Q17 presents the importance levels as Likert scale, with the S&Cs taken from the proposed model. The research assumed the importance of those S&Cs in the list were distributed in a linear increasing scale which could be rated into a scale with five response alternatives (see Table 3.5).

The five-point Likert scale was selected based on expert recommendations. First, Garland showed that using a Likert scale with a mid-point can reduce "social desirability bias" (Garland, 1991, pp. 66) in the results. To set a odd-point categories, at least three categories are needed to present the important, not important and neutral. Then, Allen argued that the scale could be increased by adding "very" to the top and bottom of the scales and "the seven-point scale was shown as the upper limits of the scale's reliability" (Allen & Seaman, 2007, pp. 64), which is consistent with the research conducted by Clason and Dormody (1994). So both five and seven are suitable for the scale. As the five-point Likert scale is a traditional classification first proposed by Likert (1932), it was chosen for this research.

In Table 3.5, all the five points indicated the increasing importance to different extent. Unbalanced scale was chosen mainly because this question was designed to help understand why each S&C was considered more or less important when estimating how important those items extracted from job postings are. A mid-point as neutral importance (neither important nor unimportant) is not appropriate for these questions, but rather a degree of importance, from low to high, is needed. They may be more likely to think out loud the reasons for the importance when they do not distracted by the unimportance. If the interviewees could not make their judgement on the importance, or namely "undecided", no importance value might be given.

In this thesis, the structure interview questions were organised by the skill categories. Participants were asked to mark the importance using numbers between 1 to 5 of their current and/or previous role.

A pilot interview was conducted to validate the interview design before conducting the full set of interviews. This resulted in some changes to the interview structure and some individual questions in the final interview. A pilot is useful for testing and revising the instruments, time arrangement and data to collect. It can also help the researcher to gain some experience in handling the interviews. However, the following issues were highlighted during the pilot:

- Participants may be not sure about the S&Cs for all roles. For example, it is common that the participant has no previous experience in test automation or there is automation tester in the companies he/she worked.

- The importance of S&Cs in the hierarchy and the logic behind them are found to be more interesting than the skill names themselves. There are already studies and investigations focusing on the importance of testers, as listed in the literature review.

So the part of structured questions was updated. Interviewees were not asked to assign importance levels for every role, but the role of their current job. Also the ten skill table was revised to tables organised by the skill categories in the proposed model (See Tables D.1 to D.6 in Appendix D).

3.3.4 Selection of Participants

It was desired to select interviewees working in organisations with a variety of different business domains. So we invited participants with this in mind. We also wanted participants to have a rich testing experience in New Zealand, so that their opinions had a solid basis for testing practice.

The potential candidates were mainly selected based on the following criteria:

- That they have more than two years' experience to make the data more convincing;
- That they work in software development organisation or company to make sure they are familiar with activities in software testing;
- That they are available in Auckland to have face-to-face interviews conveniently.

The test professionals with more experience are preferred because they know exactly what practices and techniques are used in the testing industry in NZ and are more likely to provide a holistic picture of required skills for testers in different role and seniority levels. This is because junior on junior/graduate testers may have limited recognition in the implied topics and required skills without enough experience in the industry.

A short list of suitable candidates was identified for possible participation from existing industry contacts and relationships. The combined resources of the primary researcher and the supervisor were used. The invitations were sent to potential interviewees directly.

Invitations were sent out mainly by Email, and two using WeChat [9]. The interview IDs were determined by the chronological order of reply and this ID was used to anonymise the data for subsequent analysis. We stopped sending invitations when we had ten confirmed interviews. Ten interviews were all there was time to conduct in the time available for the thesis. More than 90% of the invitations sent out were accepted. Appointments were made based on the schedule and preferences of the participants to alleviate their possible discomfort. The topic and objectives were explained as part of the invitation to participants.

3.3.5 Carrying Out Interviews

Interviews were carried out face to face in a convenient mutual place that suits the interviewee best, so that they can feel more secure and relaxed to express their thoughts during the whole process (Hammond & Wellington, 2013). 70% of the interviews were conducted near the participants' workplaces. Two were conducted in their workplaces, while the reminder one was taken in the participants' home. Half of the interviews were conducted by the leading researcher alone, while both the researcher and supervisor finished the other half. Among the interviews, two were conducted in Chinese because they can express their real thoughts better using their mother language. The research supervisor attended some of the interviews to provide support.

[9] http://www.wechat.com/en/

The same protocol was used in every interview. This can increase the reliability and uniformity of the data (Sadraei, Aurum, Beydoun & Paech, 2007). At the beginning of each interview, the origin of the research idea, the overview of research motivation and objectives were explained by interviewer, as well as the main ethics items that the participants should know. These enable the interviewee to get a general idea about the background and value of our topic. Also, they may feel more psychological safe knowing all data will be analysed anonymously. The consent Form was signed after discussing the privacy of the process and rights of participants.

During each interview, audio recordings were made using the main researcher's iPhone, while field notes were taken by both the researcher and supervisor. Afterwards, the records were scanned into the computer for backup.

The whole process was guided by the questions in the interview capture Form (see Appendix D). Most of the questions were open, so people can feel free to talk what is on their mind. Usually, key points were written down to get a good interaction with interviewees. Participants were encouraged to talk more about the importance of different skills and relationships with their working context.

After the semi-structured part of the interviews, interviewees were presented with the structured questions as a set of printed tables with a Likert scale beside each S&C from the proposed conceptual model described in the above section. They filled in each range and were asked to speak out loud what they were thinking about while they decided on the importance level they completed for each S&C. This audio was also recorded and provided insights into why the interviewee selected a particular level of importance for each S&C. Before they wrote down the importance levels, the meaning of the Likert scale values was explained.

3.3.6 Extracting and Analysing Data

Interviews aim to gain a broad understanding of the most important S&Cs from the test professionals' perspective and the levels of importance of the S&Cs.

Raw data from interviews: recordings for interviews, the notes taken during the interviews, the written importance levels, as well as notes for think-out-loud. Figure 3.7 shows a sample of the notes for one of the interviews.

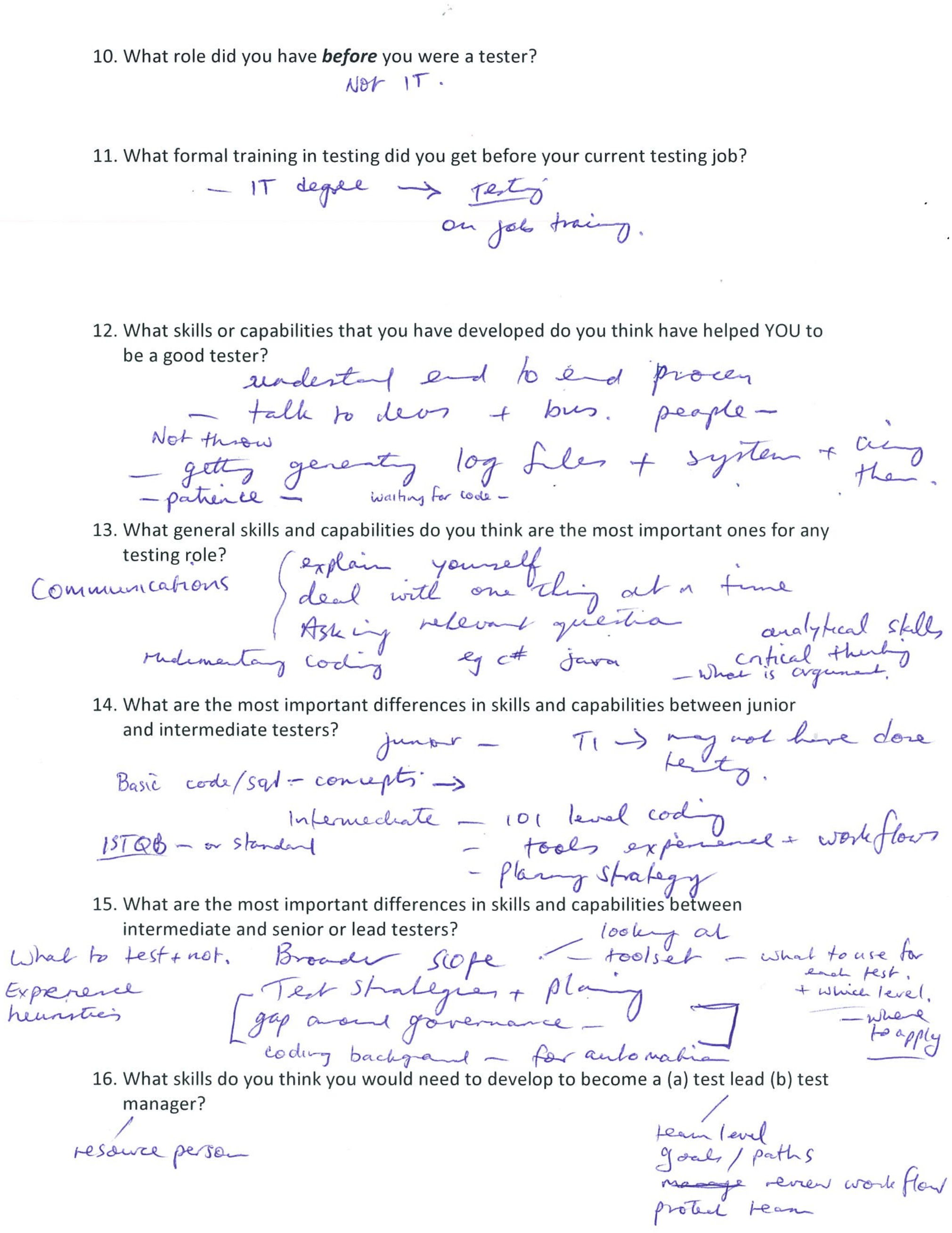
After interviews, the conversations were briefly transcribed and unnecessary information about the participants was hidden. The raw data for interviews were extracted in an Excel file. One worksheet was used to record the answers to each question, while another spreadsheet includes the Likert Scale for importance levels collected from the interviews (see Tables F.2 to F.7 in Appendix F).

All the source data were then extracted and grouped according to the design of interviews. Three different kinds of data were extracted from the interviews: answers to the semi-structured questions, assigned levels of importance for the structure questions, as well as the reasons for the importance levels.

Data from semi-structured questions were mainly analysed using thematic analysis, which is a popular approach to identify, investigate and report patterns of themes in qualitative data (Braun & Clarke, 2006). It can be used to explain different kinds of data and works well in some theoretical frameworks (Clarke & Braun, 2014). Coding using thematic analysis is actually the process of generating themes. It is widely used for interview data (Vaismoradi et al., 2013).

General stages for data extraction and classification include:

- Familiarise with source data;
- Develop initial code for each interview. Extracted data consists of the highlighted S&Cs, as well as the reasons for the importance given by test professionals;



10. What role did you have ***before*** you were a tester?

Not IT.

11. What formal training in testing did you get before your current testing job?

– IT degree → Testing
on job training.

12. What skills or capabilities that you have developed do you think have helped YOU to be a good tester?

understand end to end process
– talk to devs + bus. people –
Not throw
– getting generating log files + system + using them.
– patience – waiting for code –

13. What general skills and capabilities do you think are the most important ones for any testing role?

Communications { explain yourself
deal with one thing at a time
Asking relevant question

rudimentary coding eg c# java

analytical skills
critical thinking
– what is argument.

14. What are the most important differences in skills and capabilities between junior and intermediate testers?

junior – TI → may not have done testing.
Basic code/sql – concepts →
Intermediate – 101 level coding
– tools experience + workflows
– planning strategy
ISTQB – or standard

15. What are the most important differences in skills and capabilities between intermediate and senior or lead testers?

What to test + not.
Broader scope
looking at – toolset – what to use for each test. + which level.
– where to apply
Experience heuristics
Test strategies + planning
[gap around governance –]
coding background – for automation

16. What skills do you think you would need to develop to become a (a) test lead (b) test manager?

resource person

team level
goals / paths
~~manage~~ review workflow
protect team

Figure 3.7: An Example of Notes Taken during the Interview

- Organise data according to different themes. Interview questions were designed to map RQs;
- Analysis results and compare with findings from job adverts.

They were grouped by interview questions and a different theme associated with RQs.

Aggregation of all interviewees' views on importance levels were made in two ways: median and frequencies of importance levels. One is the median, which presents the "middle" of sorted importance levels. As the total number of the hard-copies importance levels of Likert scale is 16, so the mean of the "middle" two levels was used to present the importance level of the skill. For example, when eight forms were assigned both $\geqslant$ 4 and 5 are the middle importance levels, 4.5 is employed to represent the importance between 4 and 5, i.e. more important than 4 and less important than 5. The other measure is the frequency distribution of importance levels. The frequency of the five ordinal importance of Likert scale was counted. When analysing the data, S&Cs in the proposed model were sorted by levels of importance, from level 5 to level 1 sequentially. This presents 5 is more valuable than 4 and 4 is more valuable than 3, and so on.

The reasons why the interviewees selected the particular importance levels for each S&C were put together to gain insights into the importance of the S&Cs in the proposed model (see Table G in Appendix G).

Process of data analysis

The main analysis process includes:

- Simple analysis of demographic data;
- Comparing the most important S&Cs with those from job adverts;

- Trying to answer RQs by analysing the data based on different research questions and themes;
- Finding patterns that most participants said and interesting points that some extremely emphasised.

The analysis of background data is presented in this section, while the main findings will be elaborated in the next Chapter.

The findings from job adverts and interviews were presented and validated by two organisations and two professional networks. The feedback was positive.

3.3.7 Background Data of Interviews

The data from interviews is one of the most valuable sources for the thesis, and good selection of interviewees may strengthen the findings of interviews. So it is necessary to analyse background data of the interviews. All the demographic data can be obtained from the first part of interview questions.

Background Data of Interviewees

The roles and experience of interviewees are included in the demographic data because they may affect the objectivity of the data and can help the main researcher find patterns in the data.

Table 3.6 presents the demographic information of participants. Although P3 has an official title as Technical Test Analyst, the core responsibility is to do automation testing. So it is classified into the automation tester group.

Data for the roles and seniority levels of the participants

In terms of testing roles, 70% of the interviewees are TA, while the reminders are specialists in automation testing. This is close to the ratio of TA and AT found in job

Table 3.6: Demographic data of Interviewees

ID	Bus. Domain	seniority levels	Role Title	Dev. Process	Years of Exp.
1	Entertainment	Senior	Senior Test Engineer	Agile	6
2	Banking	Senior	Senior Test Analyst	Agile	6
3	Insurance	Senior	Senior Technical Test Analyst	Agile	>10
4	Financial Service	Senior	Senior Automation Tester	Agile	5
5	Data analytical Services	Intermediate	Intermediate QA Analyst	Agile	3
6	Payments	Test Manager	Test Manager	Agile	6
7	Media	Senior	Senior Automation Tester	Agile	>10
8	Insurance	Senior	Senior Test Analyst	Agile	>10
9	Insurance	Senior	Senior Automation Tester	Agile	8
10	Insurance	Senior	Senior Test Analyst	Agile	>10

adverts. This table also demonstrates the rich experience of test professionals who took part in the interviews. 80% are senior testers, who once worked as a junior or intermediate tester. All participants have over three years' experience in testing and 40% of them have over ten years' experience. Test Manager might be a disappearing role in agile context, especially in small companies. But the interview of TM provides us much valuable information from the top of the testing pyramid.

Job experience before testing

Q10 refers to the job history of participants before being a tester. Half of the participants had other experience before doing testing, and 40% among all of them had work experience as a developer. They all declared the experience can bring them confident in communicating with developers, understanding the technical artefacts (e.g. design of software architecture), and writing testing codes or scripts.

Formal training in testing

Q11 asks for interviewees' training history in testing. This can help us identify the importance and necessity of training or courses in testing.

Most participants learned the knowledge needed for testing on the job, rather than having formal training in testing. Some were not satisfied with the existing testing course and expressed their interest in the design of a useful practical course. Among the ten participants, only one had formal training on both theoretical and practical test, while another test professionals had a few general training on the test. P5 stated that one could learn faster in practical context; Training is needed only when there are no other test professionals in the company. P0 and P7 recommended ISTQB for basic testing knowledge.

Business domain and of organisations

Besides, the diversity of organisation and business domain can provide a broad view of the practices in NZ and make the understanding more comprehensive. In the investigations, the ten participants of interviews come from eight different organisations in seven business domain, which shows significant diversity. Table 3.7 shows the mapping between organisations and participants, while Table 3.8 illustrates the mapping of organisations and their business domain. In those tables, "O" is the abbreviation of organisation and "P" of the participant. So "O1" stands for the first organisation and "P10" the tenth participant.

Specialists in the organisations where interviewees work

Table 3.9 lists the specialist in the organisations where interviewees work at present. From this table, AT was found to be the most popular specialists, followed by PT. Some considered QA as specialist with different names, including Quality Guild/Associate,

Table 3.7: Mapping of Organisations and Participants

Organisation Number	Participants
O1	P1
O2	P2
O3	P3
O4	P4
O5	P5
O6	P6
O7	P7
O8	P8, P9, P10

Table 3.8: Mapping of Organisations and Business Domain

Organisation Number	Business Domain
O1	Entertainment
O2	Banking
O3	Insurance
O4	Financial Services
O5	Data Analytical Services
O6	Payments
O7	Media
O8	Insurance

Table 3.9: Special Roles in Different Organisations

Organisation Number	Special Roles
O1	AT, PT
O2	AT, PT, BIT
O3	AT, PT
O4	Agile Tester, QA Guild/Associate
O5	No
O6	AT, IT, PTA, QA
O7	AT
O8	QC, AT

Quality Consultant, and Quality Lead. Besides, Only O5 has no specialist.

The software development process

The Software Development Process (SDP) refers to how testers are working with, which help us to understand testers' daily work environment and analyse the importance

Table 3.10: Demographic data of Organisations

Organisation Number	Development Process	Typical Ratio of Developers and Testers	Approximate Number of Testers
O1	Scrum	3:1	30-40
O2	Scrum	5:1	>60
O3	Scrum	2:1	18
O4	Kanban	2.5:1	15
O5	Scrum	3:1	3
O6	Mixed	close to 1:1	20
O7	Scrum, Lean or Kanban	4:1 or 5:1	50
O8	Scrum, Kanban	around 2:1	50

of specific skills. The information about SDP is included in the answers of Q2-3.

All Participants declared that the development context they are working in is Agile, or changing to Agile. The most popular framework is Scrum, with two weeks for each sprint. The main development process can be described as follows: PO provides the product backlog; the development team choose features or user stories for each sprint; BA or the development team analyses the features and design the architecture; developers and testers begin to work on the project.

There were two benefits for testers working in an agile team: First, the team has a shared understanding that quality is not just the responsibility of testers, but also that of other team members; Second, testers have been involved in since the early stage of the project, and they are familiar with all artefacts. The detailed information can be found in Table 3.10.

3.3.8 Ethical Considerations

As this research refer to human participants, Ethics Application is needed. In the form, possible risks and solutions are illustrated in the application. Auckland University of Technology Ethics Committee (AUTEC) clearance was obtained for the interviews and participants involved in. The Ethics Application was approved by AUTEC on 24/07/2017 with AUTEC Reference Number 17/245. All interviewees were provided with a participant information form (attached in Appendix D). They were all asked to sign the Consent Form (attached in Appendix D), permitting the process and way to the interview, as well as the use of the information they provided in the interviews for this research. They were told to be able to withdraw the data before the data analysed. Findings will be emailed to them after produced.

3.4 Conclusion

In this chapter, the design and implementation process of the research were elaborated. The design of the study includes: identifying the RQs to guide the whole research; performing the surveys based on the analysis of data generation methods; analysing the gathered data and comparing the findings.

First, analysis on job adverts was conducted. This was carried out simultaneously with LR. To add knowledge to this topic, content blocks were identified using DCA before generating the framework of S&Cs using SCA and HC. As no comprehensive framework of S&Cs for testers was found in LR, deductive processes were used to create an initial framework of the required S&Cs. Roles and seniorities were extracted mainly from titles and the beginning part of the adverts before classified using SCA. Then, Quantitative data, i.e., occurrence frequencies of different S&Cs, were used for seeking the answers of RQs.

In addition, interviews were performed to deepen the understanding on the topic. The interviewees should be all experienced testers in NZ who are familiar with the context testers are working in NZ. Thematic analysis and qualitative analysis were mainly applied to data from semi-structured interviews. Findings from interviews were compared and synthesised with those from job adverts.

Table 3.11: Main methodological approaches used in this research

Methodological approaches	*Objective*
Literature Review	To understand the current state of knowledge of testing skills and capabilities, e.g. theoretical frameworks of competencies of testers. Also pedagogical principles from literature review will inform a testing course design, e.g. theory of course design.
Semi-structured Interview	To guide the design and conducting the interviews
Documents	To guide the implementation of job adverts
Thematic Analysis	To inform the data analysis approach for the data gathered from interviews
Summative Content Analysis Directed Content Analysis	To guide the data analysis approach for the data gathered from job advertisements

The adopted research approaches are summarised in Table 3.11.

Chapter 4

Findings and Analysis

4.1 Introduction

The chapter describes the findings from the analysis of the collected data and tries to provide the answers to the RQs.

RQ1: *What are the most important generic skills and capabilities needed by any tester, and why?*

RQ2: *What are the most important skills and capabilities that are specific for different testing roles, and why?*

RQ3: *What are the most important skills and capabilities of testers in different levels, and why?*

The whole chapter is structured by the RQs.

One of the main outputs of this thesis is a conceptual model of S&Cs, which is illustrated in Section 4.2. This model was structured as two levels. It was initially generated from the sample of job adverts and revised by the knowledge of LR and interviews. The S&Cs in the proposed model are grouped and explained by categories. Definitions are provided to avoid confusion.

The following three sections were designed to answer the three RQs respectively, both in the high-level categories and low-level S&Cs.

Section 4.3 presents findings for RQ1. Source data include job adverts, semi-structured question Q13 and structured question Q17 of interviews. Job adverts reflected the most favourable S&Cs that employers want, while answers to Q13 highlighted the most impressive S&Cs that are generic to any testers. The importance levels of the Likert scale were assigned by interviewees to answer Q17. Some important skills and capabilities in the proposed framework had not been realised by participants until answering Q17. Besides, the reasons gathered during the period of answering Q17 were used to explain why each S&C is more or less important.

Section 4.4 analyses the findings related to RQ2. Data came from job adverts, interview questions Q12 and Q17, where Q12 asked the skills that make an interviewee a good tester. In this section, different testing roles were identified by analysing the sample of job adverts and responsibilities were used to distinguish each role. Findings on each source data were triangulated from two different perspectives of employers and test professionals. The main comparison was conducted between AT and TA because they were the most popular roles representing test specialist and generalist respectively. The difference between those two roles are discussed. Then, the most important S&Cs for two roles, PTA and PT, are simply analysed.

Then, Section 4.5 elaborates the findings related to RQ3. The importance of the S&Cs for different seniority levels were obtained from job adverts and interview questions Q14-Q17. Different seniority levels identified from job adverts are analysed from the perspective of responsibilities. The most important S&Cs for each seniority level are presented and explained.

After analysing the findings, their impact on different stakeholder categories, including employers, testers, educators and researchers, are discussed in Section 4.6. Suggestions are given on how to use the findings.

4.2 Important Skills and Capabilities

4.2.1 Overview of the Proposed Model

A conceptual model of skills and capabilities was built before answering the research questions because all findings related to RQs were compared, discussed and answered base on the proposed model.

This model is one of the outputs derived mainly from the job adverts. It was initially extracted from the job adverts, and refined by knowledge gained from literature review and interviews. It is important for several reasons:

- It provides a framework for us to find the answers of the RQs and present the findings. Diverse expressions were used for different skills and capabilities in the sample of job adverts, without unified names. The proposed hierarchy of skills and capabilities can provide common names for the important S&Cs needed for testers.

- It can help potential readers to understand the findings analysed in the following section easily. A clear overview of the S&Cs needed for testers can be obtained.

- It can help test practitioners or potential testers discover the skills ignored by them.

- Employers can also think over and make a judgement on the skills they want for specific tester roles by reading through the framework.

- This model can be refined or reused by other researchers. No comprehensive and customised framework of skills and capabilities was found in the LR for testers. Most studies on this topic selected some popular items from skill sets for software developers (Matturro, 2013; Holtkamp et al., 2015).

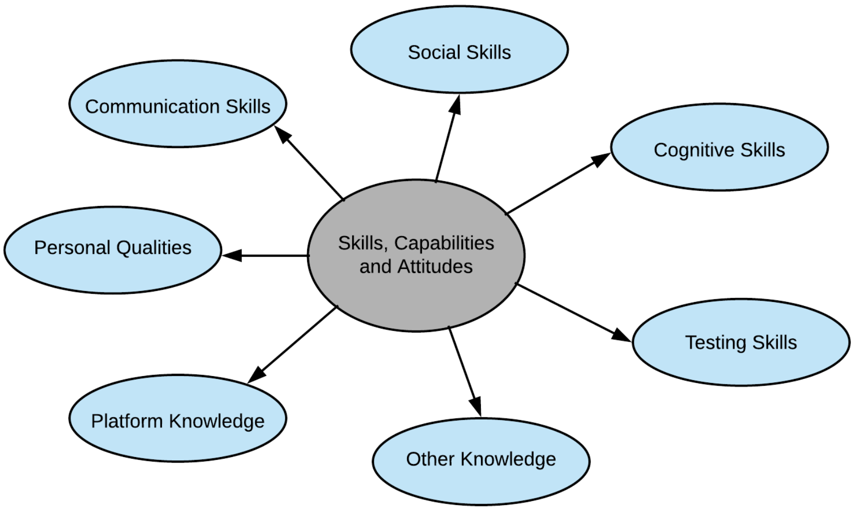


Figure 4.1: Categories in the Proposed Framework

The proposed model contains important S&Cs needed for testers and consists of two hierarchical levels.

The high level is expressed by seven categories: communication skills, social skills, cognitive skills, testing skills, other knowledge, platform knowledge and personal qualities (see Figure 4.1). Other knowledge and platform knowledge refers to knowledge on different domains, while communication skills, social skills and cognitive skills are well-known as soft skills in the literature. Personal qualities, or personal characteristics, mainly consist of individual motivation and attitudes closely relevant to testers. Some studies stated them as soft skills (Robles, 2012), while others treated them as a kind of competency (Winterton, 2009; Pawlowski & Holtkamp, 2012). Similar to capabilities, they can help improve one's performance and enable one to finish the testing job well. So, in this thesis, all personal qualities are treated as soft skills and are involved in the model. Besides, the term "skill categories" will be throughout the rest of the thesis used to mean "categories of skills and capabilities in the proposed model" unless otherwise indicated.

In the structured questions for the interviews, only six categories were employed. The items in "Platform Knowledge" were labelled as technical domain knowledge in the category "Other Knowledge" because they were determined by the business context of the software companies and it was difficult to identify the importance or prevalence of specific platform knowledge from the limited number of interviews.

The low level consists of 60 different S&Cs which were mainly extracted from the job adverts. Those S&Cs were classified by the category. Items in each category were explained or defined respectively, to provide a shared understanding on the S&Cs of the proposed conceptual model.

To present the iteration process of the model, different background colours were utilised. In the figures of the following subsections, "blue" was used as the background colour to present initial items obtained from job adverts. The items were then revised based on the knowledge obtained from the LR and the pilot interview with P0, which were presented using "Orange". After the interviews, some highlighted S&Cs that were not found in job adverts were added to the model. Those items highlighted by interviewees were marked with colour "gold".

The definitions of S&Cs in the proposed model are simply described in the following subsections, some of which were identified in LR.

4.2.2 Communication Skills

Communication Skills refer to the ability to convey information to others to be well understood and accepted, which was elaborated in Sub-section 2.4.4 in Chapter 2. Figure 4.2 shows the sub-skills related to Communication Skills. "Persuasive" and "Empathetic to others" are two abilities added to the initial model in job adverts. They play an important role when interacting with other stakeholders. Take empathetic for example. Testers are usually not so welcomed because they always tell "bad" news

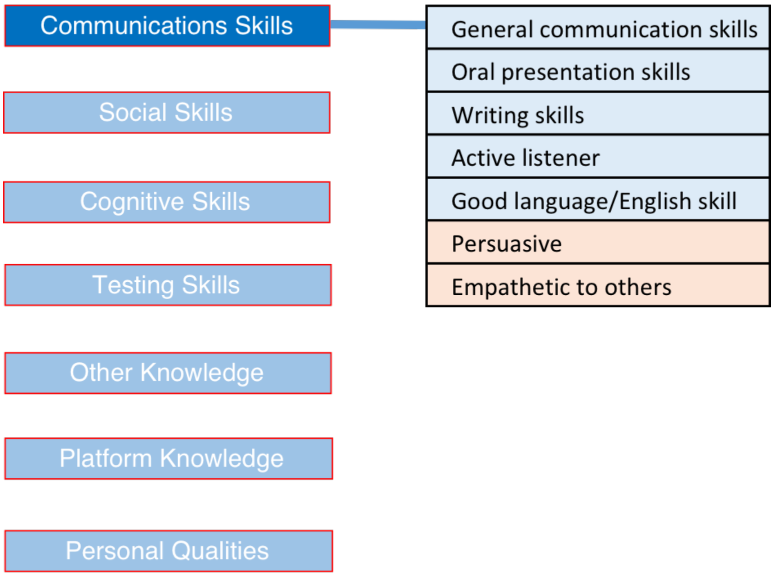


Figure 4.2: Communication Skills

before the product is ready to deploy and most developers do not like bugs in their code. However, if testers were empathetic to developers, they are more likely to convey the information to the developers: they are helping the developers improving the code quality rather than showing disappointment.

The definitions and descriptions of each skill are given as follows:

Oral presentation skills: the ability to convey your information or thoughts to others.

Writing skills: the skills to write for communication, such as writing emails, bug details, or other testing documentation.

Active listener: it refers to a tester who can concentrate, understand and respond proactively to information conveyed by others, which is kind of opposed to reflective listener.

Good language/English skill: a supporting skill for communication. English is the mother language used in NZ. But a large number of international individuals are working or trying to find work here. Language becomes a significant factor to work well in a company. So it is involved as a separate skill.

Persuasive: the ability to influence others' attitude and recognition towards something through effective oral or written communication. The subjects to persuade for testers are usually developers or other stakeholders.

Empathetic to others: the competency to feel and understand what others are experiencing. It may also mean taking into account others' personalities and cultures.

Strictly speaking, general communication skills is not a sub-skill in communication skills. However, it was used to record the number of job adverts where "communication skills" was explicitly needed in the sample. In the job adverts that asked for communication skills, around 26% used "communication skills, both oral and written" and only 5% referred to other sub-skills. The majority of job adverts asked directly for "communication skills". So, "general communication skills" was used to record the frequency of this skill category.

4.2.3 Social Skills

Social skills represent the abilities to interact with the community. They refer to any skill facilitating interactions with others and are the competencies of getting along well with others to get the job done. The main S&Cs in this category can be found in Figure 4.3.

Team Player: the ability to work effectively in a team context and contribute positively to the team goal (Ahmed, Capretz & Campbell, 2012).

Leadership: the ability to guide or manage other individuals, teams or organisations to complete tasks (John, 2009).

Ability to build relationships: it refers to build positive work relationships with colleagues, which may make the work happy and productive.

Ability to work with people at all levels of an organisation: testers may work

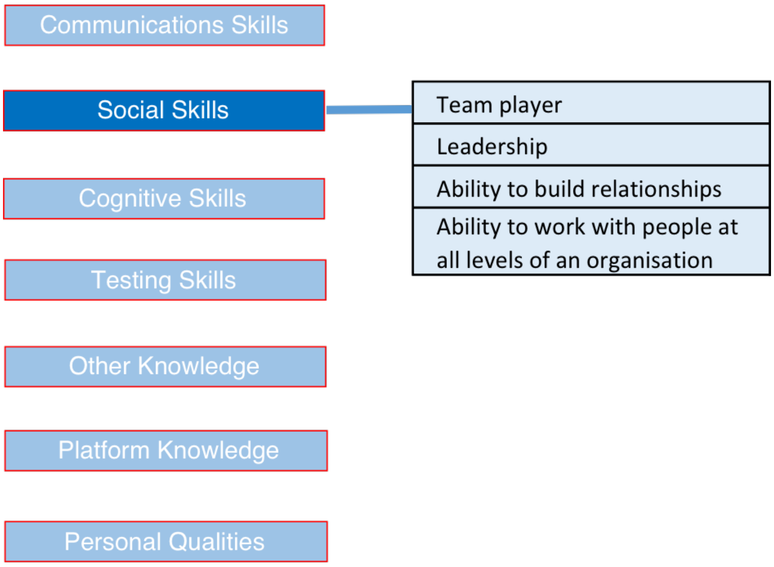


Figure 4.3: Social Skills

with different stakeholders and this competency presents your adaptability and confidence in working with different kinds of people.

4.2.4 Cognitive Skills

Cognitive Skills stand for the abilities related to mental activities, such as ways of thinking, learning and problem-solving. Those skills in the category can be found in Figure 4.4.

As mentioned in LR, there are six ordinal levels in cognitive skills: remembering, understanding, applying, analysing, evaluating and creating. All the S&Cs in this category of the proposed model belong to levels that are equal or higher than "applying".

Good judgement: the capacity to make profitable decisions by assessing or evaluating the factors and options accurately. Judgement may be made on the basis of the context and one's experience. People with good judgement can evaluate the value of different materials.

Problem solving skills: the ability to understand, analyse, and solve complex

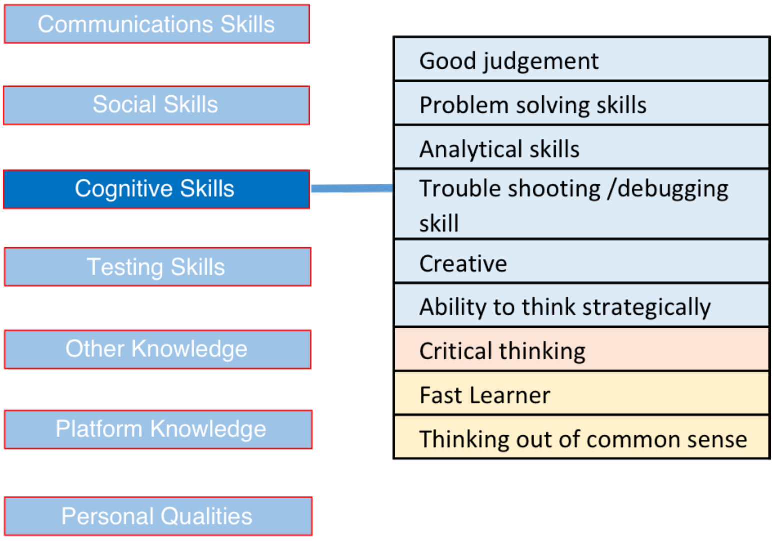


Figure 4.4: Cognitive Skills

problems and make sensible decisions based on available information (Ahmed, Capretz & Campbell, 2012).

analytical skills: it refers to the ability to discover, differentiate or organise something through scientific methods.

Troubleshooting /debugging skill: the ability to reproduce, locate or resolve a problem, commonly under the debug or development environment. Troubleshooting is sometimes associated with problem solving because it is the essential approach to solve technical issues. But for testers, this skill mainly focus on the debugging technique and process.

Creative: the ability to solve problems using creative methods (Ahmed, Capretz & Campbell, 2012).

Ability to think strategically: the ability to make effective plans and strategies align with the objectives of the company.

critical thinking: the ability to find appropriate alternatives when necessary (John, 2009).

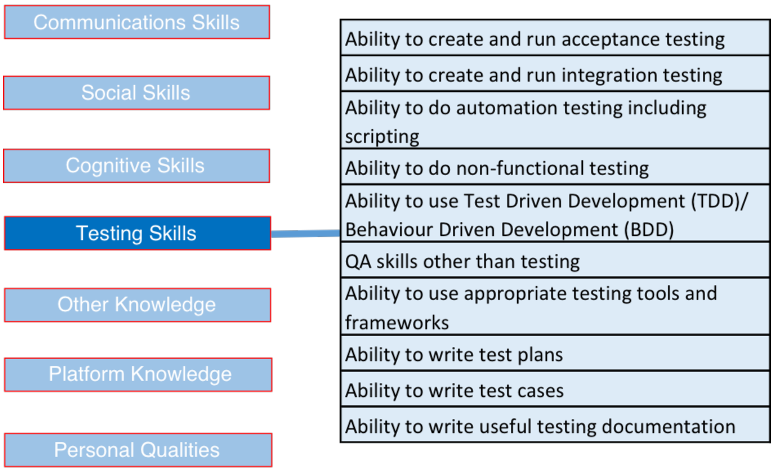

Communications Skills
Social Skills
Cognitive Skills
Testing Skills
Other Knowledge
Platform Knowledge
Personal Qualities
Ability to create and run acceptance testing
Ability to create and run integration testing
Ability to do automation testing including scripting
Ability to do non-functional testing
Ability to use Test Driven Development (TDD)/ Behaviour Driven Development (BDD)
QA skills other than testing
Ability to use appropriate testing tools and frameworks
Ability to write test plans
Ability to write test cases
Ability to write useful testing documentation

Figure 4.5: Testing Skills

Fast learner: fast learner can pick up new concepts, methodologies, and technologies quickly.

Thinking out of common sense: this means a tester can think more than what has been told and try to think out of the box, or rules.

Critical thinking and fast learner were not exposed in job adverts and were added to this framework based on a pilot interview.

4.2.5 Testing Skills

Testing Skills are closely associated with daily testing tasks and responsibilities (see Figure 4.5).

Ability to create and run acceptance tests: the ability to design and execute acceptance tests to verify the project's compliance with business requirements.

Ability to create and run integration testing: the ability to create and execute integration tests to find faults in the interaction between integrated modules of software.

Ability to do automation test including scripting: the ability to write automated test scripts or apply specific tools to automate the test process

Ability to do non-functional testing: the ability to test the non-functional requirements, such as security and performance.

Ability to use Test Driven Development (TDD)/ Behaviour Driven Development (BDD): it means a tester has the mindset and ability to design test cases and test software at the earlist stage.

QA skills other than testing: it means a tester is familiar with the QA standards and flows to assure the high quality of the software.

Ability to use appropriate testing tools and frameworks: the ability to choose and use necessary tools and test frameworks to finish testing tasks.

Ability to write test plans: the ability to plan and document the testing scope and schedules for testing activities.

Ability to write test cases: it includes the ability to create, document and execute test cases or scenarios.

Ability to write useful testing documentation: it refers to the ability to record bugs, write appropriate documents when necessary.

The skills listed in this category, such as the ability to integration testing, are closely associated with daily testing activities and tasks, or responsibilities of testers.

4.2.6 Other Knowledge

Other knowledge includes knowledge not involved in the category "testing skills". It consists of different domain knowledge, both business and technical. The S&Cs of domain knowledge are stated in Sub-section 2.4.4 and concepts are briefly explained as follows:

Business domain knowledge: knowledge related to the business logic or features of the products.

Software development process knowledge: knowledge about the main stages in

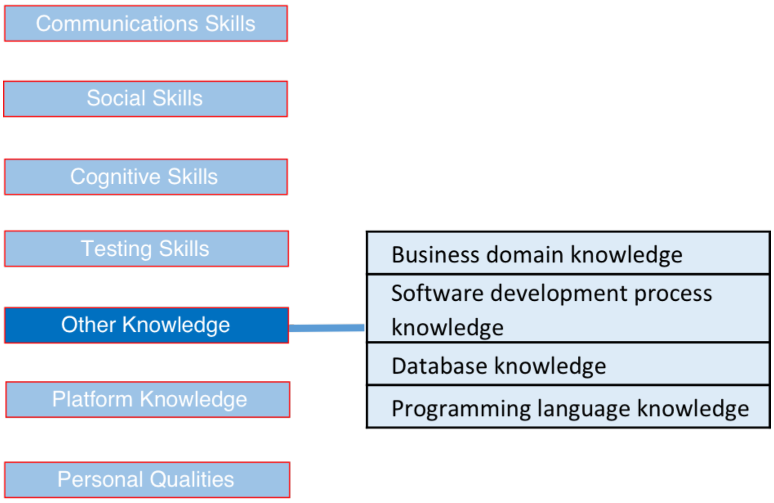


Figure 4.6: Other Knowledge

designing and developing software products, as well as the practices and principles used in the process of software development. In this research, we are interested in how testers are involved in the whole process.

Database knowledge: knowledge about different database techniques. This is an ordinary skill in many companies due to the importance of data to the business. But still, not every company needs a database.

Programming language knowledge: knowledge associated with programming or coding.

4.2.7 Platform Knowledge

As shown in Figure 4.7, platform knowledge relates to knowledge in different platforms, systems and frameworks. In the structured questions of interviews, it was replaced by technical domain knowledge.

Cloud and network infrastructure knowledge: knowledge on cloud services (e.g., AWS) and network configure.

Web domain knowledge, firmware domain knowledge and mobile domain

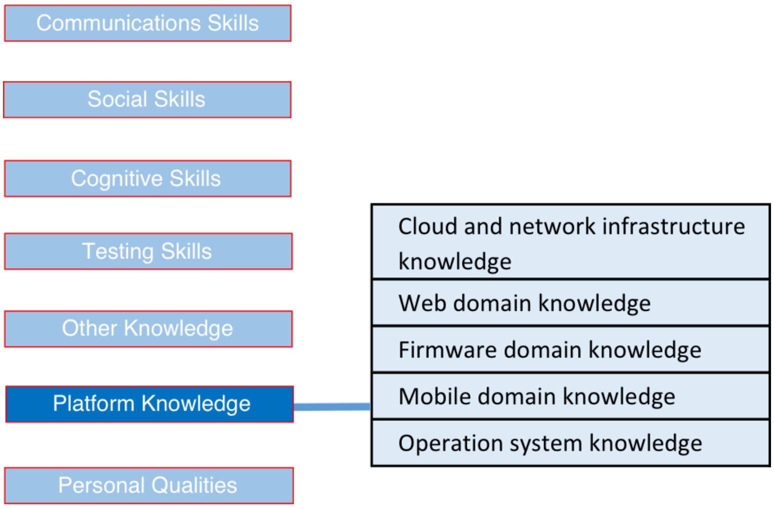


Figure 4.7: Platform Knowledge

knowledgerelated with different types of development platforms and also require different development languages and testing tools. Many apps have to support multiple platforms for various end users.

Operation system knowledge: popular operating systems include Windows, Mac, iOS, Linux systems and so on.

4.2.8 Personal Qualities

Personal Qualities, i.e. personal characteristics and attributes, are different from social skills. Many of them refer to the motivation and attitudes of individuals and were found in job adverts and literature review. They are shown in Figure 4.8 and explained in this section.

Passionate: or passion in testing, means one is quite interested in testing and enjoy the testing tasks.

Adaptable: similar to flexible. People who are adaptable are open to changes or new things when carrying out a task.

Proactive: or initiative, means to find and handle tasks positively rather than wait

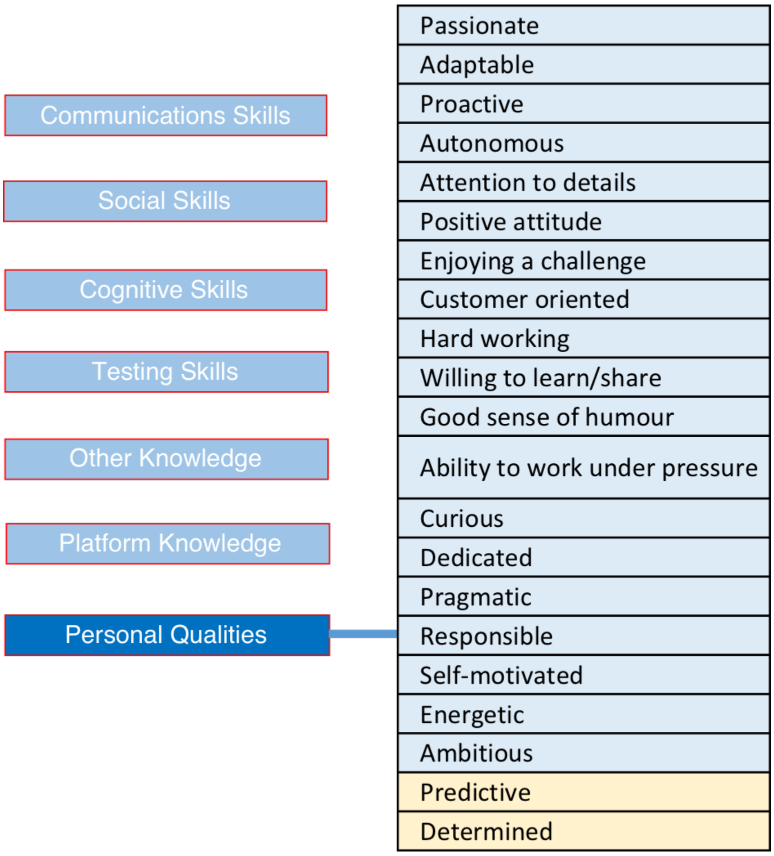


Figure 4.8: Personal Qualities

for tasks.

Autonomous: working independently to finish test tasks and find out defects as much as possible.

Attention to details: the ability to focus on small tasks in a project to discover and avoid errors.

Positive attitude: optimistic and overcoming difficulties with a productive mindset.

Enjoying a challenge: being an achiever when facing something that needs great effort to finish.

Customer oriented: equivalent to customer-focused. It means focusing on customers' long-term needs and wants.

Hard working: diligent in working or labouring.

Willing to learn/share: the desire to learn unknown things and share new techniques or knowledge with others.

Good sense of humour: relaxing people by making others laugh.

Ability to work under pressure: being able to prioritise and finish tasks due to limited time.

Curious: leveraging a passion for discovering the truth or learning unknown knowledge or new techniques.

Dedicated: dedicated people believe testing is essential and willing to spend time doing testing tasks.

Pragmatic: dealing with the problem using practical methods.

Responsible: obligate to report, explain, or justify the tasks that one needs to finish. It has similar qualities, such as accountable and commitment.

Self-motivated: knowing what to do and what needs to be done, rather than being told what to do.

Energetic: having abundance energy to do things.

Ambitious: having a desire or determination to be successful and influential.

Predictive: having the intuition to identify high risk blocks of the project and find defects.

Determined: determined testers will not change their mind or opinions easily and have confident even when others disagree with them.

Predictive and determined are two abilities emphasised by some participants of interviews.

4.3 Generic Tester Skills and Capabilities

4.3.1 Introduction

This section aims to explore the possible answers to RQ1: the general S&Cs required for any tester. It is useful to understand the specific S&Cs that are perceived as being of great importance for *any* tester, regardless of their role or seniority. Software testing as a technical occupation has particular requirements to finish the tasks well.

In this section, the data related to RQ1 are extracted and analysed. Three data sources may answer RQ1: job adverts, data associated with interview questions Q13 and Q17. The findings in job adverts provided the most frequently mentioned S&Cs from the perspective of the adverts writers, who are likely to be a Test Manager, someone from Human Resources (HR) department or someone from an employment agency. The data from the interviews were gathered using both semi-structured questions as well as structured questions based on Likert scale. The importance of each of the S&Cs identified from the job advert were assigned by interviewees using Likert scales. Interview questions Q13 was directly designed to answer RQ1 from the perspective of test practitioners. The findings from Q13 reflected the most impressive and valuable S&Cs to those test professionals in interviews. Q17 refers to the structured questions about the importance levels of Likert-type. The reasons for the importance of specific S&Cs were gathered when test practitioners were writing the importance levels. The explanations on frequencies of importance levels and the reasons for the importance were used to understand why the S&Cs in the proposed model were more, or less, important.

4.3.2 Important Skills and Capabilities for Any Testers

The analysis of findings in this subsection provides an overall understanding to RQ1. The findings from both job adverts and interviews were synthesised after discussion. As the proposed model consists of two hierarchical levels, data can be analysed in both high-level categories and low-level S&Cs.

High-level Categories

The importance of each category in the proposed model was evaluated by the weights of their frequencies. Let $f_1, f_2, ..., f_7$ be the count of job adverts that contain any item in Category 1, 2, ..., 7 respectively, which denotes the frequency of occurrence. Then, w_i, the weight of Category i, can be calculated by the following equation:

$$w_i = \frac{f_i}{\sum_{i=1}^{n} f_i}$$

.

Figure 4.9 illustrates the weights of the seven skills categories for any tester, which gives us a common idea of employers' demand when seeking a tester. A regular polygon in the radar diagram stands for an axis with the same weights and the radar values are 0%, 5%, ..., 20% respectively. This figure shows that *testing skills* and *other knowledge* are significantly highlighted. The weights of frequencies are around 50% more than those of communication skills, personal qualities and platform knowledge.

Tockey argued that hiring managers seeking software engineers prefer to ask for learn-able technologies in their job postings (Tockey, 2015), which adds weight to the findings of this research.

In the sample of job adverts, recruiters seemed to focus more on candidates' knowledge and experience for completing the testing tasks, or namely technical skills, than other categories. Testers with knowledge or experience in the similar context were

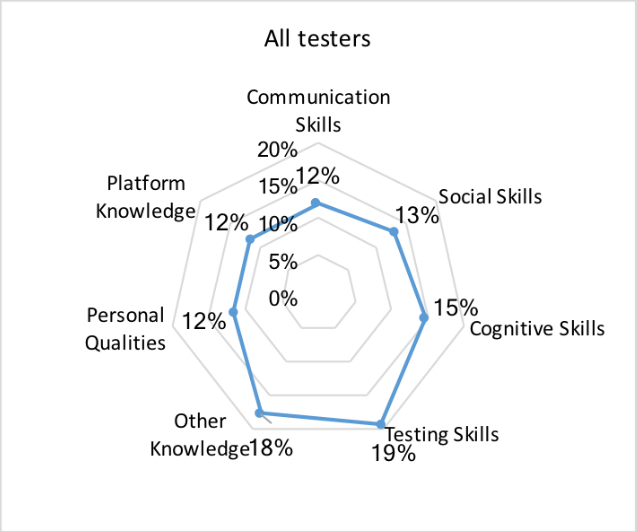


Figure 4.9: The Weights of Skills Categories for Any Tester from Job Adverts

expected to be more competent, or skilled, in finishing the test tasks.

However, interviewees argued that the importance of them depends on the context (practices, tools and techniques) of a specific company. The knowledge varies that important knowledge or skills needed in this company may not be useful in another. Besides, the testing techniques updates very fast and testers can never know everything when they join the company and

Instead, test professionals in the interviews highlighted personal qualities, which may determine how good the tester can be. Participants stated that those personal qualities might help testers easily to master some skills in a short period and finish testing tasks in high quality.

The importance of communication skills was widely recognised by interviewees, which is consistent with that finding in the LR. Testers with excellent communication skills can win the trust of other stakeholders, including clients, developers and product owners. Also, they can help testers understand what developers are doing and have done, and make decisions on where to test. Developers are more likely to give priority to fix bugs when well communicated. In the past, one may finish the testing tasks without talking too much with others. But nowadays, especially in an agile team, testers may have various meetings every day. The tester has to communicate frequently with the team. P2 emphasised the importance of communication skills like this: "A person with 70% communication skills and 30% technical skills will prompt faster than one with 30% communication skills and 70% technical skills."

Interviewees also emphasised the importance of having a broad overview of different technical knowledge: it helps testers to understand the developers' "language" and interact effectively. Technical knowledge, not limited to the those listed in platform knowledge mentioned in the proposed model, can help testers think more comprehensively and strategically. Testers with sound technical background may find that it is easier to understand kinds of artefacts than those with no or little knowledge of the techniques.

Low-level Skills and Capabilities for Any Tester

The previous analysis provided us with an initial view of the distribution of different categories. However, bias might exist due to the unbalanced number of various items of each category in the proposed model. So, it is necessary to analyse the specific S&Cs in the low level. In this thesis, the top ten S&Cs were analysed unless otherwise indicated. Sometimes, there were more than ten items because the last few items had the same frequency.

To get more details, the most important S&Cs from each source data were analysed to find answers to RQ1.

Figure 4.10 shows the top ten S&Cs for all testers based on the frequency and its percentage of the S&Cs in the proposed model. The frequency and its percentage were consistent because the total number of job adverts is constant. The data in the figure refers to the percentage of job adverts that includes particular skills or capabilities. The detailed frequency of all S&Cs is shown in Table F.1 in Appendix F.

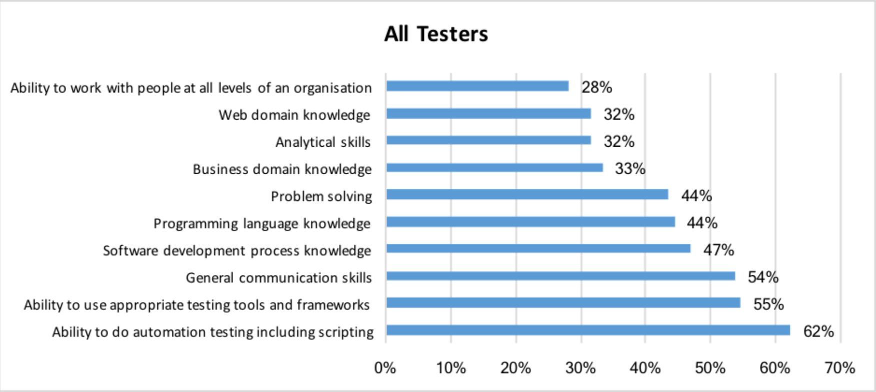


Figure 4.10: The Top Skills and Capabilities of Any Tester Found in Job Adverts

Table 4.1: Highlighted Skills and Capabilities Obtained from Q13 of Interviews

Fast learner
Attention to details
Analytical skills
Willing to learn/share
Active listener
Asking questions
Thinking out of common sense
Predictive
Determined

Interview questions Q13 was designed to discover the most important S&Cs that were generic to any testers. Table 4.1 lists the S&Cs highlighted in the Q13. "Highlight" has two different meanings here: either with high frequency in interviews, which denoted it was mentioned by more interviewees, or emphasised by one or more participant.

To triangulate the findings from different perspectives, the findings from job adverts, interviews and LR were compared using a Venn diagram. Figure 4.11 shows the most important S&Cs found in three different sources. Almost all the highly-ranked S&Cs in LR were included in the sample of job adverts. The S&Cs found in sample of job adverts are more similar to the findings of LR, compared with those of interviews.

Besides, two indicators, frequency distribution and median, were used to identify the importance levels of S&Cs in interviews. The frequency distributions, averages and

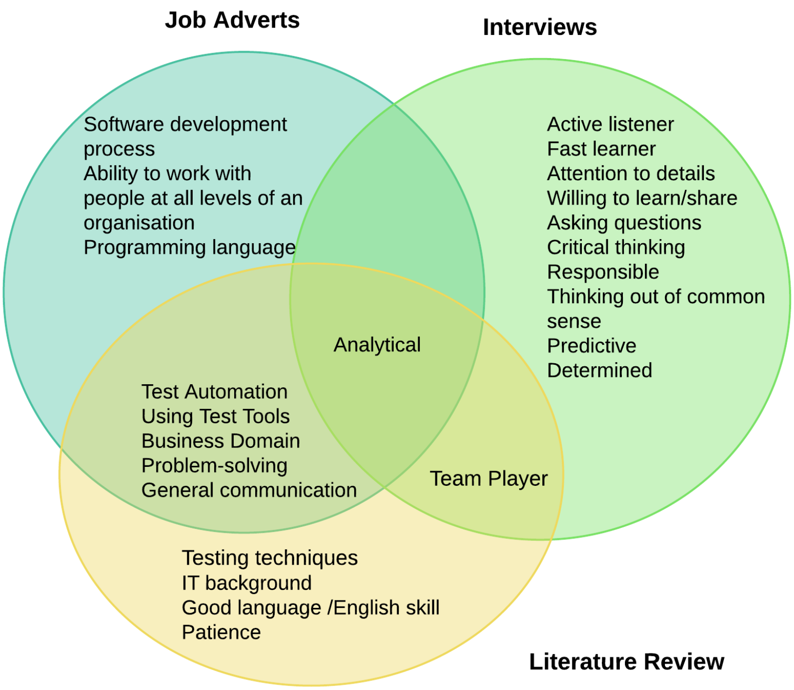


Figure 4.11: Answers for RQ1 from Different Sources

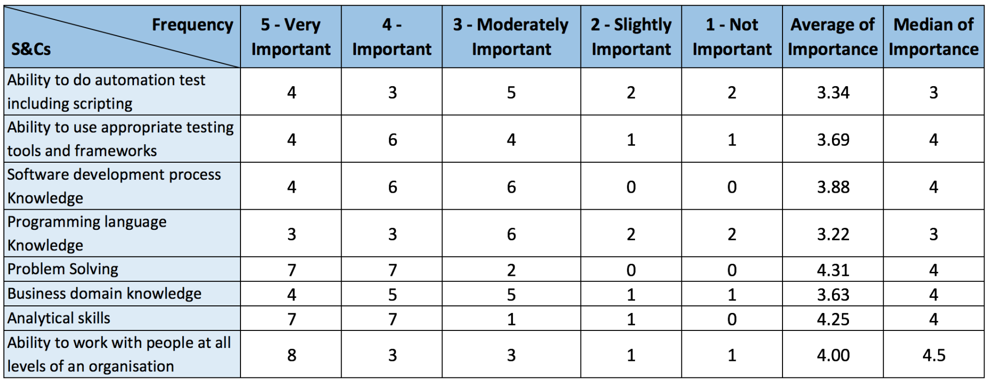
medians of importance levels for those top S&Cs identified by job adverts and interview questions Q13 are presented in Tables 4.2 and 4.3 respectively. Averages and medians was introduced to help provide an intuitive impression of the importance levels.

There was a total of 16 importance levels collected. This figure is grouped by the ordinal Likert scales of importance levels. Among all the top S&Cs, "willing to learn/share" wins the most "Level 5" in the interviews, which reflects the open attitude to keep on learning new techniques and grow with the team.

The following presents the reasons why a specific skill or capability is more, or less, important. The top S&Cs from job adverts are first analysed, followed by those obtained from the interviews.

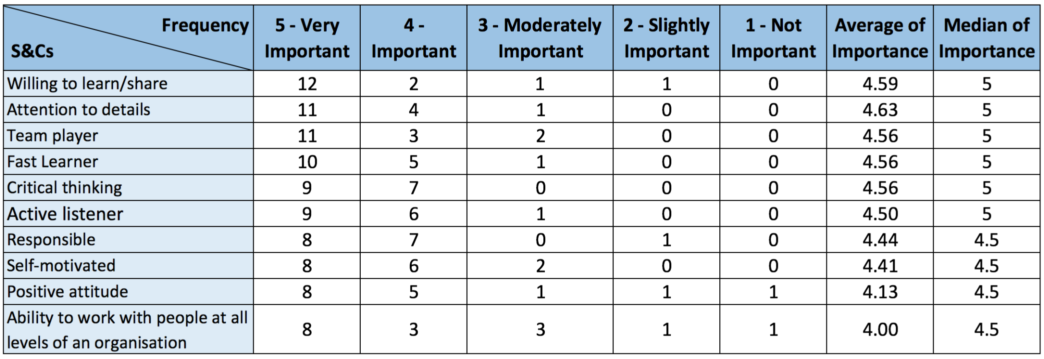
Ability to do automation testing including scripting: The ability to do automation testing is most frequently mentioned in job adverts. The results reflect their close

Table 4.2: The Frequencies of Importance Levels for the Top Ten Skills and Capabilities of Any Tester Obtained from Job Adverts



Frequency S&Cs	5 - Very Important	4 - Important	3 - Moderately Important	2 - Slightly Important	1 - Not Important	Average of Importance	Median of Importance
Ability to do automation test including scripting	4	3	5	2	2	3.34	3
Ability to use appropriate testing tools and frameworks	4	6	4	1	1	3.69	4
Software development process Knowledge	4	6	6	0	0	3.88	4
Programming language Knowledge	3	3	6	2	2	3.22	3
Problem Solving	7	7	2	0	0	4.31	4
Business domain knowledge	4	5	5	1	1	3.63	4
Analytical skills	7	7	1	1	0	4.25	4
Ability to work with people at all levels of an organisation	8	3	3	1	1	4.00	4.5

Table 4.3: The Top Ten Skills and Capabilities Ordered by Importance Levels from Q13



Frequency S&Cs	5 - Very Important	4 - Important	3 - Moderately Important	2 - Slightly Important	1 - Not Important	Average of Importance	Median of Importance
Willing to learn/share	12	2	1	1	0	4.59	5
Attention to details	11	4	1	0	0	4.63	5
Team player	11	3	2	0	0	4.56	5
Fast Learner	10	5	1	0	0	4.56	5
Critical thinking	9	7	0	0	0	4.56	5
Active listener	9	6	1	0	0	4.50	5
Responsible	8	7	0	1	0	4.44	4.5
Self-motivated	8	6	2	0	0	4.41	4.5
Positive attitude	8	5	1	1	1	4.13	4.5
Ability to work with people at all levels of an organisation	8	3	3	1	1	4.00	4.5

relationship with the daily tasks of testers. Automation testing is becoming prevalence in modern testing. It is also a sign of the strong demand and market trend in the expertise of those. The market vacancies indicate the short and mid-term gap between the demand and supply to some extent. The majority interviewees state that it is moderately important or above. However, less than half of the participants think they are important. According to the median, interviewees identify it "moderately important". The proponents believe it is daily work for automation testers. But one participant argued that it is hard to define the importance for any tester because an agile team is diverse and they have different expertise.

Ability to use appropriate testing tools and frameworks: Testing tools and frameworks are widely utilised in various activities, such as performance testing, test

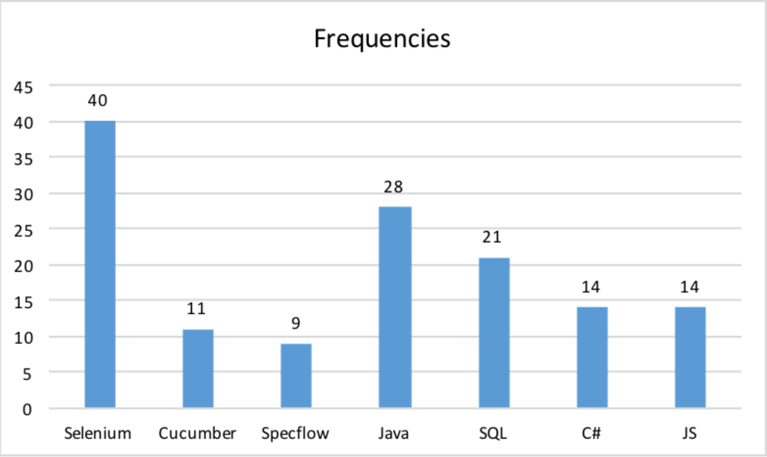


Figure 4.12: Frequencies of Popular Tools and Programming Languages

cases management, bug tracking, etc. According to data from interviews, ten of 16 have importance level ⩾ 4. only two results showed this ability to be less than moderately important.

The frequencies of tools and programming languages in the job adverts is illustrated in Figure 4.12. It shows that selenium, a representative tool for automation test, was well accepted by employers.

Software development process knowledge: The knowledge is relative to the team practices. According to the collected sample of job adverts, agile, the most common practice in the software industry of NZ, was explicitly mentioned in around half of them. The experience of working in a similar team is widely concerned by employers. All test professionals in interviews think it is above average importance. According to the collected data, this is important mainly because it helps you work well in a team, especially in an agile environment. Knowledge of the workflow and process can help testers to gain a shared understanding with their team members and collaborate well with them. Besides, the tester, as a member of the team, also has to think what is working or not in your organisation, and how to improve it. This is especially important in agile teams.

Programming language knowledge: This knowledge frequently appears in the sample of job adverts. Figure 4.12 shows java is the most popular language mentioned in the sample of job adverts. Interviewees have different opinions for this item. Six of 16 results marked it to be moderately important. But it still has strong supporters. In the collected data, four interviewees had experience as a developer and declared that programming language knowledge contributes positively to their testing jobs. They stated the knowledge on this area greatly helps them to become a good tester. As many testers are not good at programming, those with previous development experience are more likely to know the thoughts of developers and the principles of design. They can easily understand the developers' language and communicate well with them. Also, one can track the cause of some defects if he or she can read the code and use some debugging skills. For instance, P5 benefits from the developer's experience to great extent because he can show the developers the cause of the bugs.

Problem solving: It is emphasised both in the sample of job adverts and the LR. The importance of problem solving is ignored at the stage of answering interview questions. But when marking the Likert Scale of the items, no participants denied the importance of it. 14 forms marked the importance $\geqslant 4$, while 7 of them consider this skill very important. Some supporters noted that testers might confront with various problems in different levels and they cannot ask others for help every time. So, it is valuable to have the ability.

Business domain knowledge: One-third of the sample of job adverts asked the knowledge on business domain. It is marked as one of the top competencies in most studies found in LR. The majority interviewees considered them moderately important at least, while only four of them felt it very important. According to the conversations in interviews, the importance is evident. It can help testers design test cases and do acceptance test. If a tester has a whole view of the domain, he or she may have an advantage in finding some defects that others do not know. They are more likely to predict which part may hide the most bugs and make plans for the modification of code. In contrast, if a tester is not familiar with the domain knowledge, he/she may have some bias on understanding and may miss critical bugs. Besides, it can also help do non-functional testing, especially performance test. P3 denoted the data for a load test and stress test relies heavily on the business context.

Analytical skills: This item is marked by high importance in all sources. In the grid form of interviews, 14 are marked important and seven very important. Importance is widely discussed by interviewees. They can help review the artefacts and find bugs. Testers should know how to analyse the results and "make sure a bug is a bug" (P1). Also, they can help explain the impact of changes that developers made and make test plans. There are many decisions that testers need to make, such as what to be tested first, what is affected by the modifications, and what is the cause of specific defects. These kinds of decisions rely on the tester's analytical skills. Besides, they can help testers understand the architecture and code so that testers can talk to developers in their language.

Ability to work with people at all levels of an organisation: Although it is at the end of the top list in sample of job adverts, the interview data indicates the importance of this skill is between 4 and 5. Eight participants mark it "very important". The reason for the importance is that testers work in a diverse team. Testers have to interact with other people rather than keep on clicking in front of the computer. But some interviewees argued that some testers, especially juniors, do not have the opportunity to contact people at different levels.

Attention to details: The importance of this skill is confirmed by the majority of interviewees. The importance levels of this skill were assigned to "very important" in twelve forms. This is a generic skill that testers must have and the importance is based on the recognition that testing is to discover defects in different artefacts during the development process. It can help testers find bugs and locate the root causes efficiently and effectively. Also, testers could know what to test if they were details oriented.

Willing to learn/share: This is a precious quality. Learning and sharing make one grow to be a better tester by gaining a broad view of the newest techniques and methodologies. "It the key part of tester to grow," said P4.

Team player: The importance of team player was elaborated in the LR. From the perspective of test practitioners in the interviews, all testers work in a cross-functional team, and even the test manager is part of a team. Team player helps them work collaboratively with developers and other stakeholders. Besides, the tester is a supporting role that can help improve team performance and productivity. They should be a good team player to help developers write high-quality code.

Fast learner: Interviewees had different opinions on fast learner. Many advocates recommended fast learner due to the ever-updating techniques. Only one opponent stated that "fast learner do not make you a good tester". "When you confront with some problem, fast learner doesn't help." Another participant argued that it is more important to ask for help when you meet some difficulty.

Critical thinking: It is emphasised by more than half of interviewees. It helps testers to generate their own views on different issues, rather than believing whatever developers said. It decides whether one can grow up to senior testers to some extent. Just like what P4 said: "This is the baseline skill for senior tester".

Active listener: According to the interviewees, testers should be a good listener to get enough information from stakeholders and win their trust. First, listening to others can help testers to get enough information before taking actions. For example, what developers are doing, what is the way they are thinking, and what they are not sure about. Besides, it is a good way to win the trust of others because people are more likely to feel understood.

Responsible: It is the work ethics an employee should have. Testers have to responsible for both the team and the company (P6).

Asking questions: It can be considered as a kind of critical thinking that is of great importance in real work. It is different from the oral presentation skills in communication skills which refers to the art of asking. Because asking questions means that you know what exactly you want to ask and what kind of information could help. It is a kind of thinking that matters more. For a general testers, it is more important to ask what you want to know from developers or other stakeholders. Techniques are changing very fast and you can follow up all the technical knowledge, testers have to focus on what is related to the high quality of their product and what other team members, especially developers, are doing. It is usually faster and more accurate to know issues, such as the progress, the architect and the design of the projects, than read the code and documents by yourself. Just as P8 said, you should have a bright idea of what is exactly you want to know and that you cannot figure out in short time. Of course, you should also know who to ask for help and when. According to what P5 said, you can save a lot of time and energy to find the right person to ask. In a development team, especially in an agile team, people are good at different areas. So most of the time, you can find a person to answer your question. But everybody has their own tasks, so you should also choose the right time to ask questions.

Thinking out of common sense: Four participants emphasised the importance of this skill. Testing is not just limited to bug discovery or breaking the software system down by particular operations. Testers have to think more than daily testing. They need to think from the "real" user's perspective and think over the specifications and design. A broad view of the whole system and concepts are expected.

Predictive: This skill is mentioned a few times by testers. Professional testers usually have some intuition of which modules may include most problems. They can easily identify the highly risky areas and design test cases for them. This not only helps write test plans, but also benefits the exploratory testing. Testers always do not have enough time to test everything. This is especially useful in an agile team due to the limited time frame for continuous deployment (CD).

Determined: This skill is emphasised by P2. Think about the following scenario: You found a potential defect. But when you told the developer, they denied your thought, "oh, no, it is not a bug", or "it is not a big deal, we are not going to change it in this version". What will you do in this scenario? Good testers will feel confident about their findings and insist their own opinions even if the developers disagree.

4.3.3 Top Generic Skills and Capabilities in Each Category for Any Testers

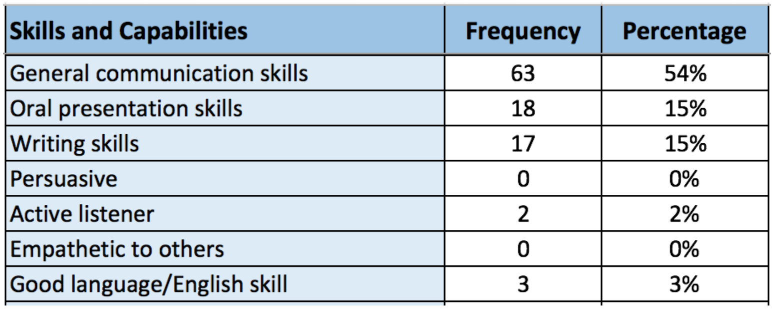
It is meaningful to understand the top S&Cs in each category. For example, this can give people a shared understanding on the most important skills in each category. When talking about each category, there should be some representative skill that demonstrates the importance. For example, when one states social skills are important for testers, people will realise that:" Yeah, tester should be a team player and work with people at all levels of an organisation". Also, Educators can focus on these most important skills when they design a course.

In this section, comparisons will be performed from both perspectives of job adverts and test professionals. Original data obtained from interviews can be found in Tables F.2 to F.7 in Appendix F, where the S&Cs with importance levels over four are highlighted.

Top Skills and Capabilities in Communication Skills

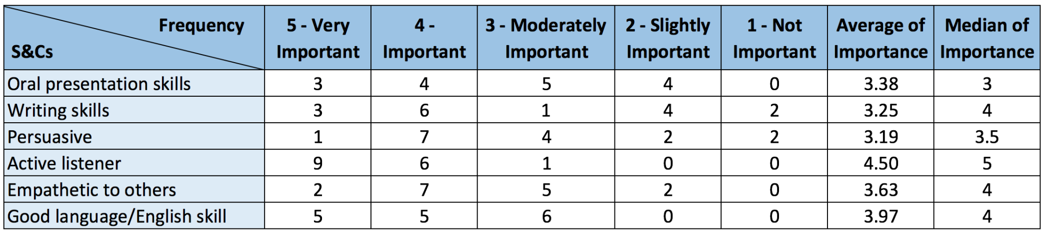
In job adverts, 54% emphasised general communications skills (see Table 4.4), which is the same as the finding from LR. About one-quarter of them using "good communication skills, both oral and written". From the sample of job adverts, recruiters knew the importance of communication skills, but it was not clear what aspects of communication skills were most important.

Table 4.4: Frequency and Percentage of Communication Skills in Job Adverts



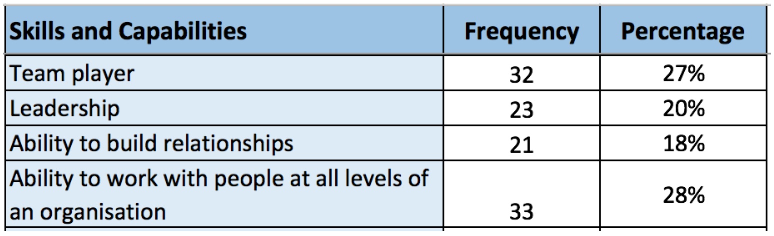
Skills and Capabilities	Frequency	Percentage
General communication skills	63	54%
Oral presentation skills	18	15%
Writing skills	17	15%
Persuasive	0	0%
Active listener	2	2%
Empathetic to others	0	0%
Good language/English skill	3	3%

Table 4.5: Frequency and Percentage of Communication Skills in Interviews



Frequency S&Cs	5 - Very Important	4 - Important	3 - Moderately Important	2 - Slightly Important	1 - Not Important	Average of Importance	Median of Importance
Oral presentation skills	3	4	5	4	0	3.38	3
Writing skills	3	6	1	4	2	3.25	4
Persuasive	1	7	4	2	2	3.19	3.5
Active listener	9	6	1	0	0	4.50	5
Empathetic to others	2	7	5	2	0	3.63	4
Good language/English skill	5	5	6	0	0	3.97	4

Table 4.6: Frequency and Percentage of Social Skills in Job Adverts



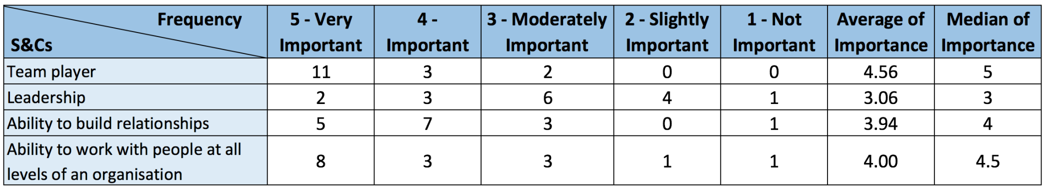
Skills and Capabilities	Frequency	Percentage
Team player	32	27%
Leadership	23	20%
Ability to build relationships	21	18%
Ability to work with people at all levels of an organisation	33	28%

However, data from the interviews showed "active listener" was the most important communication skill. Table 4.5 shows how "active listener" stands out from the skills.

Top Skills and Capabilities in Social Skills

Table 4.6 shows the frequency and percentage of social skills, where the frequencies of "team player" and "ability to work with people at all levels of an organisation" were almost the same. Table 4.7 illustrates the frequency distribution of social skills. According to the interview data, "Team player" was recognised as one of the most generic skills to work well in a team by interviewees.

Table 4.7: Importance Levels of Social Skills in Interviews



Frequency S&Cs	5 - Very Important	4 - Important	3 - Moderately Important	2 - Slightly Important	1 - Not Important	Average of Importance	Median of Importance
Team player	11	3	2	0	0	4.56	5
Leadership	2	3	6	4	1	3.06	3
Ability to build relationships	5	7	3	0	1	3.94	4
Ability to work with people at all levels of an organisation	8	3	3	1	1	4.00	4.5

Top Skills and Capabilities in Cognitive Skills

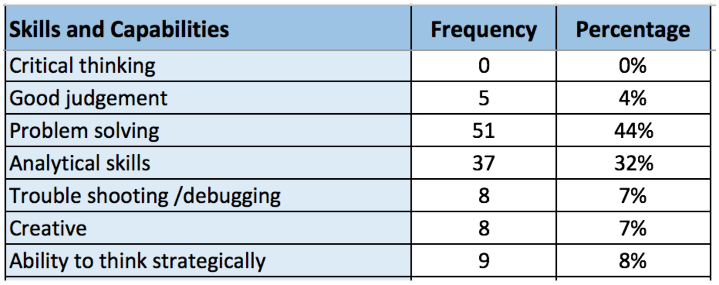
Tables 4.8 and 4.9 are data associated with the importance of S&Cs in cognitive skills. Top skills found in job adverts are problem-solving skills and analytical skills, where problem-solving has much higher frequencies than analytical skills. The importance was confirmed by the importance levels obtained from interviews. Fourteen answers out of 16 identified them as being "important" and "very important". However, one participant did not think problem-solving was important for any tester, especially for juniors. Another one stated that it is the developer's responsibility to solve problems, while testers are responsible for discovering the defects. Besides, those two skills were discussed together in LR.

Although problem-solving and analytical skills are important, critical thinking and being a fast learner are confirmed by more interviewees. All interviewees agreed that critical thinking was important and nine answered are scaled as "very important". Actually, all the cognitive skills, except creative, are considered as "important" (level 4) by the majority of participants. This may indicate the importance of cognitive skills for testers.

Top Skills and Capabilities in Testing Skills

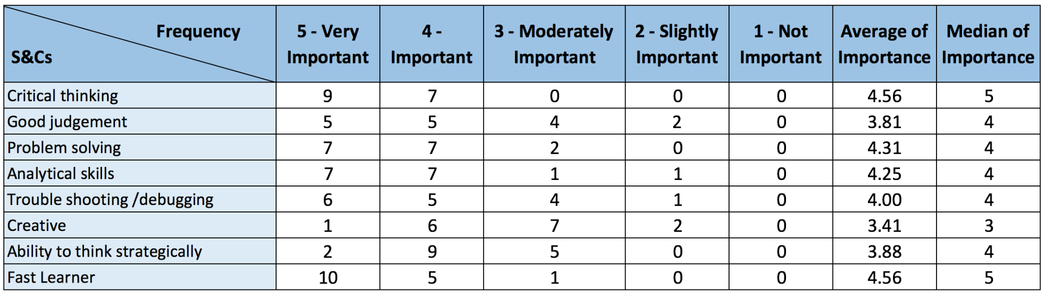
Tables 4.10 and 4.11 are data associated with importance of S&Cs in testing skills. In job adverts, automation testing and related testing tools were most frequently mentioned. However, the median of the importance levels were "moderately important" and even

Table 4.8: Frequency and Percentage of Cognitive Skills in Job Adverts



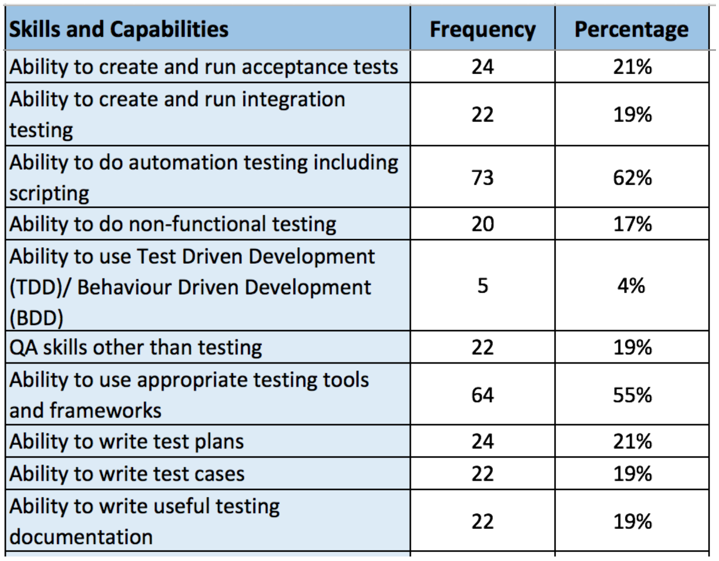
Skills and Capabilities	Frequency	Percentage
Critical thinking	0	0%
Good judgement	5	4%
Problem solving	51	44%
Analytical skills	37	32%
Trouble shooting /debugging	8	7%
Creative	8	7%
Ability to think strategically	9	8%

Table 4.9: Importance Levels of Cognitive Skills in Interviews



Frequency S&Cs	5 - Very Important	4 - Important	3 - Moderately Important	2 - Slightly Important	1 - Not Important	Average of Importance	Median of Importance
Critical thinking	9	7	0	0	0	4.56	5
Good judgement	5	5	4	2	0	3.81	4
Problem solving	7	7	2	0	0	4.31	4
Analytical skills	7	7	1	1	0	4.25	4
Trouble shooting /debugging	6	5	4	1	0	4.00	4
Creative	1	6	7	2	0	3.41	3
Ability to think strategically	2	9	5	0	0	3.88	4
Fast Learner	10	5	1	0	0	4.56	5

Table 4.10: Frequency and Percentage of Testing Skills in Job Adverts

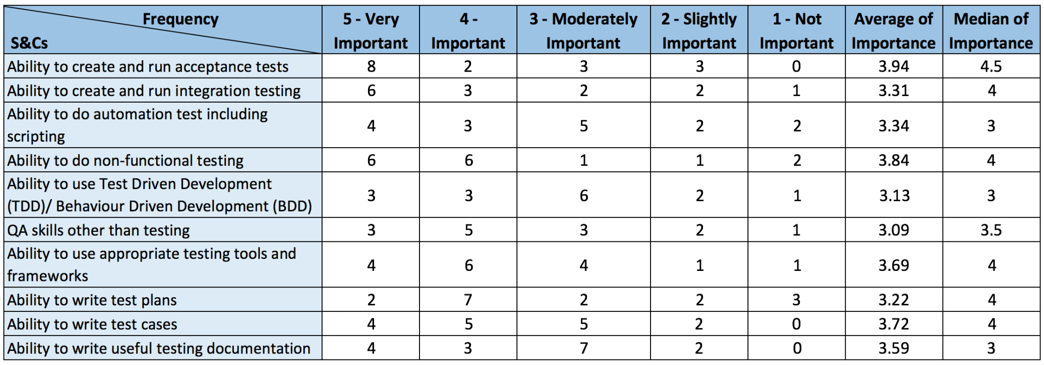


Skills and Capabilities	Frequency	Percentage
Ability to create and run acceptance tests	24	21%
Ability to create and run integration testing	22	19%
Ability to do automation testing including scripting	73	62%
Ability to do non-functional testing	20	17%
Ability to use Test Driven Development (TDD)/ Behaviour Driven Development (BDD)	5	4%
QA skills other than testing	22	19%
Ability to use appropriate testing tools and frameworks	64	55%
Ability to write test plans	24	21%
Ability to write test cases	22	19%
Ability to write useful testing documentation	22	19%

two argued that they were not important at all. In contrast, the abilities to do acceptance testing and non-functional testing are recognised as more important.

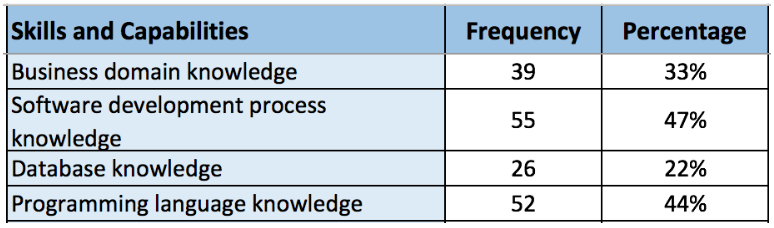
Interviewees stated that the majority of testing skills were not so important, although they were preliminary requirements and daily work for testers. They argued that some

Table 4.11: Importance Levels of Testing Skills in Interviews



Frequency S&Cs	5 - Very Important	4 - Important	3 - Moderately Important	2 - Slightly Important	1 - Not Important	Average of Importance	Median of Importance
Ability to create and run acceptance tests	8	2	3	3	0	3.94	4.5
Ability to create and run integration testing	6	3	2	2	1	3.31	4
Ability to do automation test including scripting	4	3	5	2	2	3.34	3
Ability to do non-functional testing	6	6	1	1	2	3.84	4
Ability to use Test Driven Development (TDD)/ Behaviour Driven Development (BDD)	3	3	6	2	1	3.13	3
QA skills other than testing	3	5	3	2	1	3.09	3.5
Ability to use appropriate testing tools and frameworks	4	6	4	1	1	3.69	4
Ability to write test plans	2	7	2	2	3	3.22	4
Ability to write test cases	4	5	5	2	0	3.72	4
Ability to write useful testing documentation	4	3	7	2	0	3.59	3

Table 4.12: Frequency and Percentage of Other Knowledge in Job Adverts

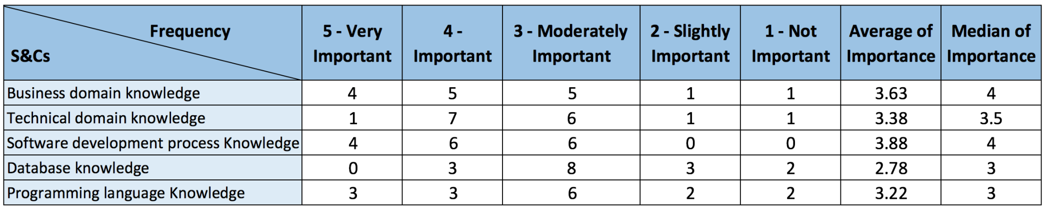


Skills and Capabilities	Frequency	Percentage
Business domain knowledge	39	33%
Software development process knowledge	55	47%
Database knowledge	26	22%
Programming language knowledge	52	44%

skills, such as abilities to write test plans, test cases, and test documentation were not so important anymore. Little documentation is needed, and also, different templates are provided for writing.

The abilities to do acceptance testing and non-functional testing were two most important S&Cs identified by interviewees. They were marked important by 10 of 16 answers. Doing acceptance testing gained higher attention because testers have to think from the users' point of view and make sure all the requirements and specifications are done properly. Meanwhile, non-functional testing is important for almost every company because performance and security of their products are two special aspects of great importance. For example, performance may affect users' experience and the system may collapse under certain loads if not well considered and tested before deployment.

Table 4.13: Importance Levels of Other Knowledge in Interviews



Frequency S&Cs	5 - Very Important	4 - Important	3 - Moderately Important	2 - Slightly Important	1 - Not Important	Average of Importance	Median of Importance
Business domain knowledge	4	5	5	1	1	3.63	4
Technical domain knowledge	1	7	6	1	1	3.38	3.5
Software development process Knowledge	4	6	6	0	0	3.88	4
Database knowledge	0	3	8	3	2	2.78	3
Programming language Knowledge	3	3	6	2	2	3.22	3

Top Skills and Capabilities in Other Knowledge

The skill determined by different sources were the same in this category: software development process knowledge. No interviewees thought it was slightly important or not important.

Top Skills and Capabilities in Personal Qualities

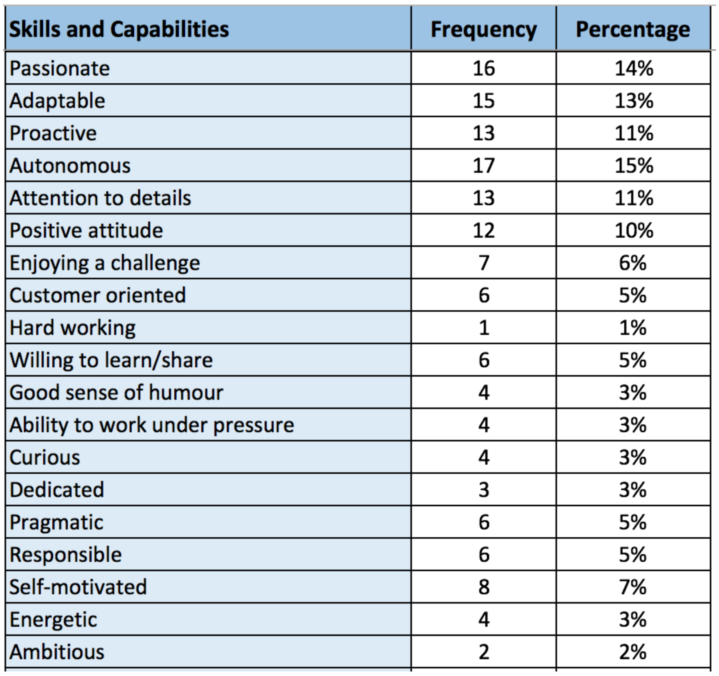
Table 4.14 illustrates the frequency and percentage of skills in personal qualities. The skill list for personal qualities is the longest in the model, while the frequencies of those skills found in job adverts were the least. People who wrote the adverts did not pay much attention to personal qualities in job adverts. However, several personal qualities were highlighted by interviewees (see Table 4.15).

Considering the frequency distribution of importance levels, "attention to details", "willing to learn/share", "responsible", "self-motivated" and "positive attitude" can be marked as "very important", while "pragmatic", "adaptable", "proactive", "curious" and "custom oriented" can be marked as "important". They all have the labels with no less than half of the sum.

Responsible: basic work ethics to testers. Interviewees argued that every tester has assigned tasks and responsibilities and they should always be responsible for both the team and the company.

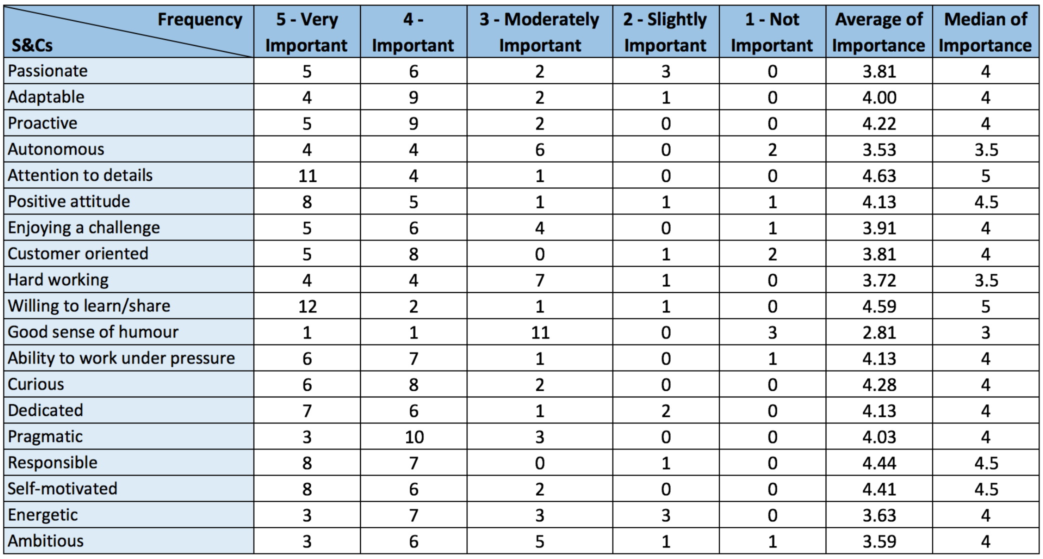
Passionate: According to the participants, one may become a tester and finish different testing tasks without passion, but passion will make one an excellent tester

Table 4.14: Frequency and Percentage of Personal Qualities in Job Adverts



Skills and Capabilities	Frequency	Percentage
Passionate	16	14%
Adaptable	15	13%
Proactive	13	11%
Autonomous	17	15%
Attention to details	13	11%
Positive attitude	12	10%
Enjoying a challenge	7	6%
Customer oriented	6	5%
Hard working	1	1%
Willing to learn/share	6	5%
Good sense of humour	4	3%
Ability to work under pressure	4	3%
Curious	4	3%
Dedicated	3	3%
Pragmatic	6	5%
Responsible	6	5%
Self-motivated	8	7%
Energetic	4	3%
Ambitious	2	2%

Table 4.15: Importance Levels of Personal Qualities in Interviews



Frequency S&Cs	5 - Very Important	4 - Important	3 - Moderately Important	2 - Slightly Important	1 - Not Important	Average of Importance	Median of Importance
Passionate	5	6	2	3	0	3.81	4
Adaptable	4	9	2	1	0	4.00	4
Proactive	5	9	2	0	0	4.22	4
Autonomous	4	4	6	0	2	3.53	3.5
Attention to details	11	4	1	0	0	4.63	5
Positive attitude	8	5	1	1	1	4.13	4.5
Enjoying a challenge	5	6	4	0	1	3.91	4
Customer oriented	5	8	0	1	2	3.81	4
Hard working	4	4	7	1	0	3.72	3.5
Willing to learn/share	12	2	1	1	0	4.59	5
Good sense of humour	1	1	11	0	3	2.81	3
Ability to work under pressure	6	7	1	0	1	4.13	4
Curious	6	8	2	0	0	4.28	4
Dedicated	7	6	1	2	0	4.13	4
Pragmatic	3	10	3	0	0	4.03	4
Responsible	8	7	0	1	0	4.44	4.5
Self-motivated	8	6	2	0	0	4.41	4.5
Energetic	3	7	3	3	0	3.63	4
Ambitious	3	6	5	1	1	3.59	4

who finish tasks well. Because it can drive one to understand the principles behind the testing activities and make an improvement.

Adaptable: Based on the interview data, "adaptable" had two different kinds of understanding: to work well in a context with different culture and personality, or

Table 4.16: Comparison of Top Skills in Each Category

Categories	Top skills and capabilities in job adverts	Top-ranked in grid forms of interviews
Communication Skills	General communication skills;	Active listener;
Social Skills	Team player; Ability to work with people at all levels of an organisation	Team player;

Cognitive Skills	Problem-solving skills;
	Analytical skills;
		Critical thinking;
		Fast learner;

Testing Skills	Ability to do automation test including scripting; Ability to use appropriate testing tools and frameworks;	Ability to create and run acceptance tests; Ability to do non-functional testing;
Other Knowledge	Software development process knowledge;	Software development process knowledge;

Personal Qualities	Autonomous; Passionate; Adaptable;	Attention to details;
		Willing to learn/share;
		Responsible;

flexible to finish different testing tasks. The importance was based on the diversity of working context and ever-changing techniques.

Autonomous: It was considered as the basic work ethic that was extremely to senior testers.

Discussion

Table 4.16 compared the top S&Cs in each category from job adverts and answers to structured questions of the interviews. In the table, data of the second column is obtained from the frequencies of specific S&Cs in the proposed model while that of the third column origins from the frequency distribution of the importance levels using Likert Scale.

4.4 Skills and Capabilities for Different Role

This section aims to explore the possible answers to RQ2: the most important S&Cs for each testing role. So the roles should first be identified before finding answers to RQ2.

The different role has particular requirements on S&Cs. The source data related to RQ2 include the sample of job adverts, answers to interview question Q12 and the structured questions of interviews. Q12 refers to what S&Cs makes the interviewee a good tester and the answers can reflect the most valuable S&Cs needed for their current role. Besides, limited by the number of interviews, the frequency distribution will not be used to discuss the importance levels of the S&Cs. There were only five TA, four AT and one TM were interviewed.

As many top skills found for some roles have been explained in Section 4.3, the reasons for the importance will not be repeated. This section will focus on the roles found in collected data and the different S&Cs for those roles.

4.4.1 Tester Roles

Generally speaking, testing roles can be divided into two major categories: Generalist and Specialist. A specialist is a skilled tester who is responsible for specific types of testing or using special techniques, compared to generalists.

Tester was the most general name used to refer to an individual doing testing activities in the sample of job adverts. A preferable name for general tester was test analyst (TA), which was employed to represent the non-specialist. Sometimes, it was called Software Test Engineer (STE). Test Lead and manager were considered as high-level testers and were grouped into this category. In this thesis, some samples, such as A11, A34 and B29, had the word QA or quality in the title, but were classified as TA because they had quite similar responsibilities with TA and experience of TA role was expected in many job adverts for QA. The recruiter merely emphasised the importance of quality to products. In this thesis, TA was defined as a generalist. The main responsibilities of TA include ordinary testing activities, such as writing test plans, designing and executing testing cases, and analysing test results.

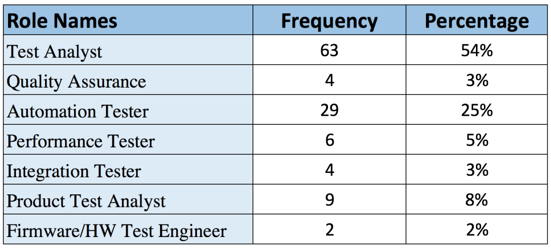
As "test analyst" or "tester" were quite common terms in the sample of job adverts and they were widely used with other roles. For example, having combined with term "automation", i.e., "Automation Test Analyst" or "Automation Tester", they became specialists. AT is a test engineer role, which focuses on software test automation and technical testing strategies. Other synonyms in job titles, such as automation test engineer, automation test analyst, automation test lead, and automation expert, were also categorised into TA.

Similarly, PA is the role, responsible for load and performance testing. Performance is an important indicator of software quality, which determines the importance of performance testing. P1 said:" We do not have special titles for different roles. All called test engineer combining different roles for everybody. But we do have a separate role for performance tester." However, according to the interviews, not every company had a separate role for PT at that time. Sometimes, AT was expected to do the tasks as PT.

In the job advert sample, two job adverts for Technical Test Analyst (TTA) were found. One was seeking a TTA who could be responsible for both automation testing and performance testing. We discovered that AT was also responsible for performance testing in several job adverts, such as B36; their primary responsibility was to conduct automation test. Thus, they were labelled as AT. The other TTA was just responsible for performance testing and then was marked as PT.

QA, another generalist, is defined as the tester who focus on the quality issues, such as setting up the quality standard, improving quality processes, or make sure the products reach the quality standard. They are responsible for the quality of the product by software testing or other ways. According to the sample of job adverts and interview

Table 4.17: Classification of Roles Found in Job Adverts



Role Names	Frequency	Percentage
Test Analyst	63	54%
Quality Assurance	4	3%
Automation Tester	29	25%
Performance Tester	6	5%
Integration Tester	4	3%
Product Test Analyst	9	8%
Firmware/HW Test Engineer	2	2%

data, it may be called "Quality Analyst", "Quality Engineer", or "Quality Coach" in NZ industry. It is worth mentioning that some "Quality Test Analyst" was found to do the same job as TA and "Quality" was just a preference name that denotes their attention to quality. These samples were recognised as TA, rather than QA. This was verified from the interviewees. In some organisations, QA sometimes was called "QA Guild" or "QA Associate".

InT refers to a tester who is responsible for integration testing. Integration testing is the testing executed after unit testing and can be done either manually or automatically. Through the advert samples, most integration testing was actually done by automation testers. Besides, only four adverts from the sample were seeking a specialist in integration testing, however.

Besides, Firmware/HW TE are not studied due to the difficulty to find enough samples. This was not a traditional software test role and was very rare in job adverts in NZ market. In the collected data, there were only two samples and they seemed to come from the same company according to the description of the job. No job adverts on this role was found in the following three months.

Table 4.17 shows the classification and the frequency of testing roles obtained from the sample of job adverts. According to the sample of job adverts, at that specific period, TA was the most common role needed in the NZ market, while AT was the most frequently mentioned specialist.

4.4.2 Skills and Capabilities for AT and TA

The analysis on the difference in responsibilities can help clarify the requirements on S&Cs for different roles. So at the beginning of this part, the duties obtained from job adverts are presented.

TA and AT can be considered two most common roles in the NZ market. Their differences also reflect the different characteristics of generalist and specialist to some extent. So the differences between AT and TA are discussed. After that, a simple analysis on PT and PTA is performed. Due to the limited size of job adverts seeking PT and PTA, no conclusions were obtained.

Difference in Responsibilities

AT and TA are two different roles. The required S&Cs for testers are determined by their responsibilities to large an extent because testers benefit from their exceptional skills to finish different testing tasks effectively. Figure 4.13 illustrates the most frequently mentioned responsibilities of AT and TA. The frequency was obtained by counting the adverts that contain the responsibility for this role. As the number of job adverts for AT and TA was not the same, they were normalised by dividing the size of sample for the specific role.

To make the difference more clear, Table 4.18 presents the contrast based on the rank of responsibilities between TA and AT obtained from job adverts.

The most apparent difference between TA and AT can be summarised as follows:

- Although automation testing is becoming the top responsibility emphasised in job recruiting for both roles, AT has higher frequency ratios on both the ability to do automation testing including scripting and to use tools and frameworks than TA.
- TA is expected more to create and run acceptance tests, while AT focuses on the ability to do non-functional testing".

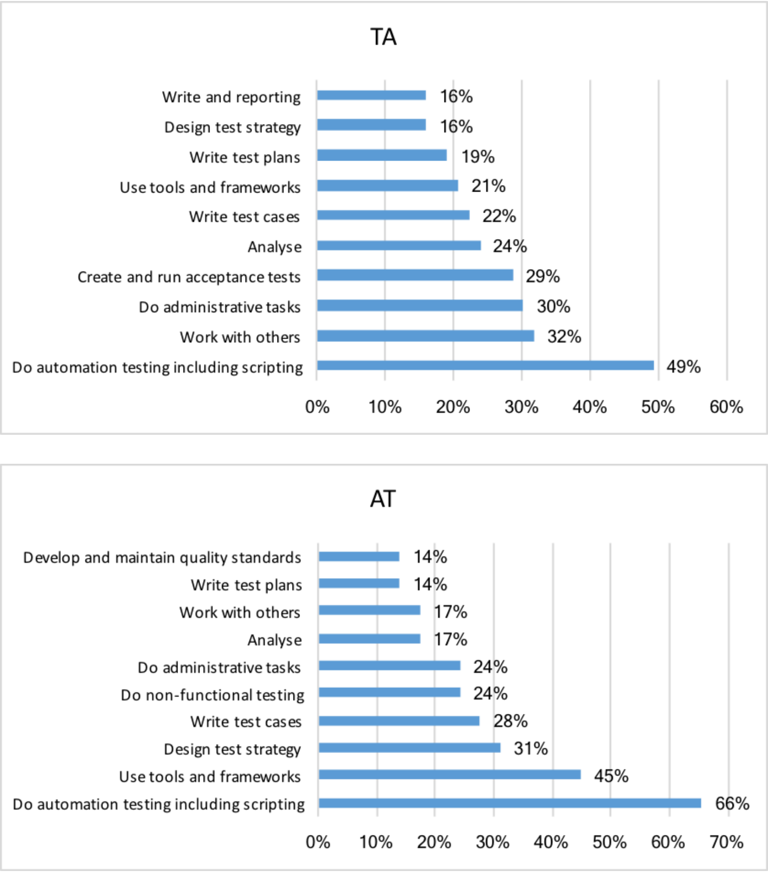


Figure 4.13: The Responsibilities for TA and AT

- TA takes more responsibilities on working with others or doing administrative tasks than AT, while AT is expected to do more on designing test strategy.

Table 4.19 lists the main differences of responsibilities between AT and TA.

Difference in Top Skills and Capabilities

By aggregating the quantitative data, expected skill categories for different roles were captured from a macroscopic point of view. Figure 4.14 shows the weights of skill categories for the different roles, which was calculated the same as the methods described in Section 4.3.2. In this figure, the weights of the required skill categories for both TA and AT are similar. One difference is found: TA needed more communication skills while AT required more testing skills in the sample of job adverts.

Table 4.18: Rank of Top Responsibilities for AT and TA

AT	Responsibilities	TA
1	Do automation testing including scripting	1
2	Use tools and frameworks	7
3	Design test strategy	9
4	Write test cases	6
5	Do non-functional testing	13
6	Administrative	3
7	Analyse	5
8	Work with stakeholders	2
9	Write test plans	8
10	Develop and maintain quality standards	11
11	Create and run acceptance tests	4

Table 4.19: Main Differences of Responsibilities for AT and TA

	AT	TA
1	Do automation testing including scripting Use tools and frameworks	Work with others Analyse
2	Do non-functional testing	Create and run acceptance test
3	Design test strategy	Administrative

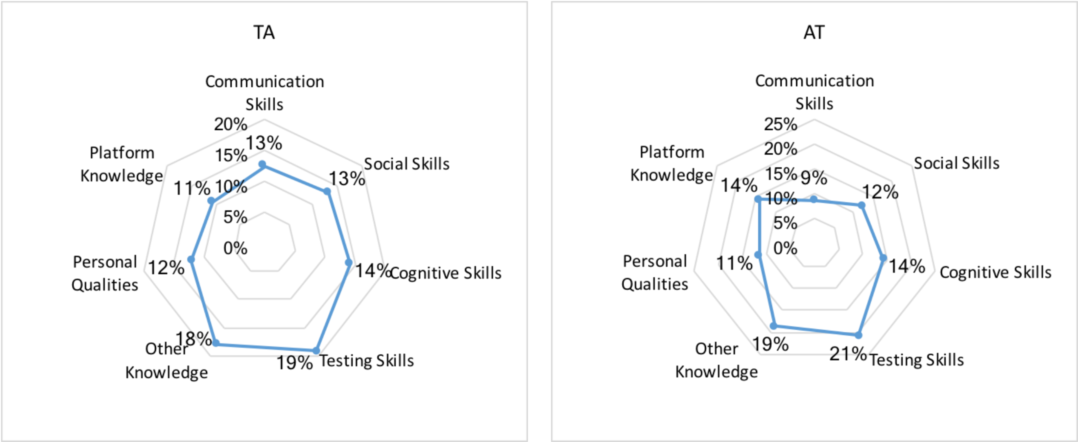


Figure 4.14: The Weights of Skills Categories for TA and AT

Then, the required S&Cs are analysed. Figure 4.15 shows top 10 S&Cs required by AT and TA. It shows almost all the percentages for AT are higher than those for TA. However, a significant overlapping of the top items (70%) was observed, including "Analytical skills", ‘Web domain knowledge’, "Programming language knowledge",

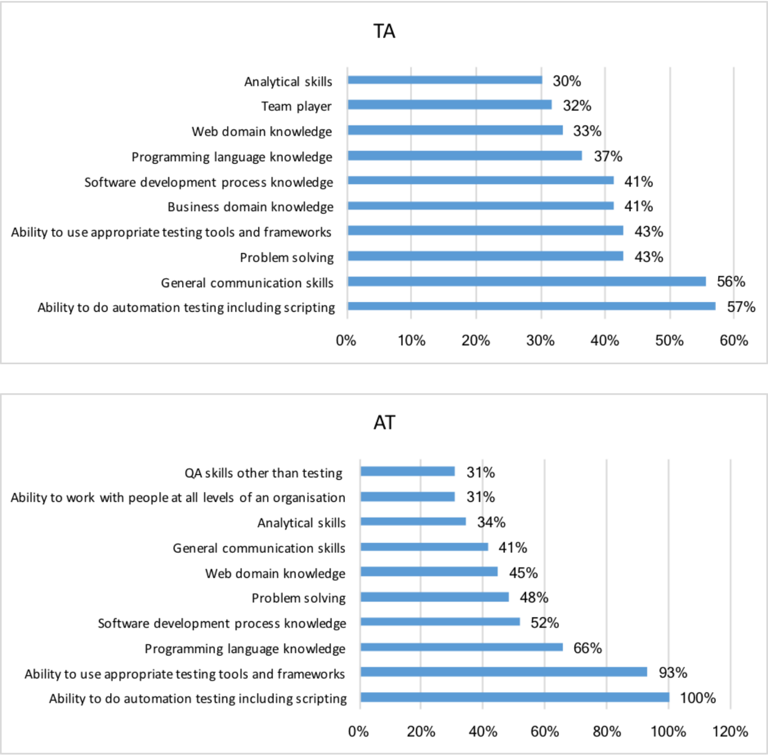


Figure 4.15: The Top Skills for TA and AT

"Software development process knowledge", "Ability use appropriate testing tools and frameworks", "Problem solving" and "Ability to do automation testing including scripting".

AT is a specialist in test automation. From the data of job adverts, this role seems to have higher requirements than TA on the following S&Cs: "ability to do automation testing including scripting", "ability to use appropriate testing tools and frameworks", and "programming language knowledge".

It is interesting that nearly 60% TAs also have the requirement of doing automation testing. This shows the popularity and employers' expectation of doing automation testing. P9 declared that "one may still find a job on manual testing, but automation testing is a trend". If one has enough knowledge on programming language and testing tools, there may be more job opportunities on testing.

For TA, two items have higher values than AT. First is general communication skill.

Business domain knowledge is quite important for TA. It can help TA reduce the bias on understanding the requirements and specifications. However, the importance is shifting to code knowledge (P8). This can be seen from the increasing importance of programming language knowledge.

Just like the skills for any tester, the most important S&Cs for AT and TA are quite different from those obtained from job adverts.

One of the most significant differences between them are "self-motivated" for AT and "curious" for TA. AT is a technical testing role. The reason for the importance of self-motivated may be the fast updating technologies. According to what P3 said, ATs have to update their knowledge database every three months. A self-motivated AT is more likely to keep up with the changing of techniques and do the test better. Curiosity is of top importance for ATs because one of their responsibilities is to discover defects by exploration testing. Curiosity enables testers to explore defects more proactively.

Table 4.20 shows the highlighted S&Cs collected through interview question Q12. This table lists the highlighted S&Cs for both TA and AT, followed by the frequency of each of them. During the interviews, technical domain knowledge was emphasised by four test analysts, while the ability to understand code was highlighted by three Automation Testers (ATs). This reflects the significant similarity and difference between AT and TA. ATs in the interviews argued that understanding code is different from "programming language knowledge". Testers do not need to know the knowledge on programming languages to write code as the developers. What they need is to understand what developers have done or not. As a TA, a broad view of technical domain knowledge is enough to handle daily testing tasks. By contrast, an AT is responsible for test automation and should understand how to code and discuss with developers in technical ways. Besides, communication skills and willingness to learn were also identified as most valuable S&Cs for TA.

Table 4.20: Highlighted Skills and Capabilities related to Interview Question Q12

Role names	Number of role	Highlighted S&Cs and their frequencies
TA	6	Technical domain knowledge (4); communication skills (3); willing to learn (3); faster learner (2); attention to detail (2); analytical skills (2); problem solving skills (1); collaborative (1); pationate (1); patience (1); know what to do (1); asking questions (1); curious (1); think out of common sense (1); determined (1); persuasive (1);
AT	4	Ability to understand code (3); fast learner (2); predictive (2); attention to details (1); problem solving (1); passionate (1); patience (1); ability to do automation testing (1); business domain knowledge (1); know what to do (1); persuasive (1);

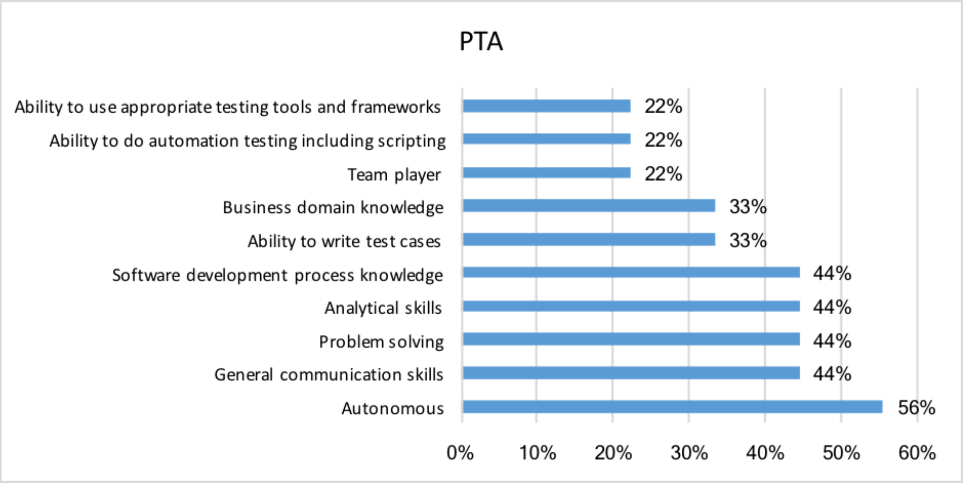


Figure 4.16: The Top Ten Skills and Capabilities for PTA from Job Adverts

4.4.3 Simple Analysis on PTA and PT

PTA

PTA is a special testing role. Through the collected data, four job adverts in nine had no description of responsibilities. So the collected responsibilities were not so representative.

The S&Cs needed for PTA was analysed directly. Figure 4.16 illustrates the most important S&Cs needed for PTA.

The top skills required for PTA is work autonomously. Most employers expect that

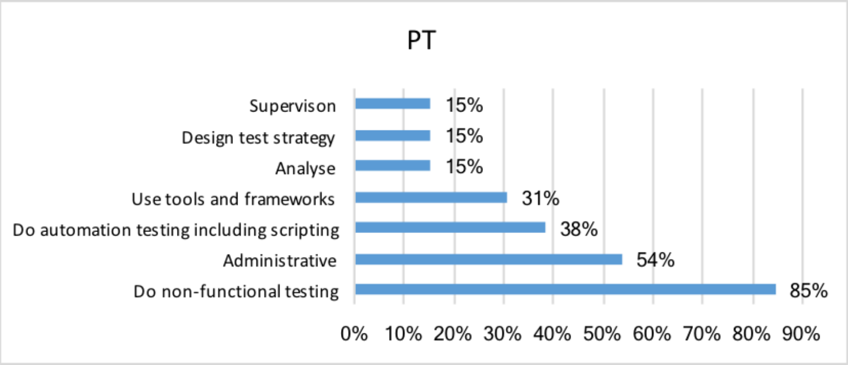


Figure 4.17: Extracted Responsibilities of PT

PTA should be familiar with the testing techniques and can complete tasks independently. The interviewees argued that autonomy is a fundamental work ethics for all testers in different levels and increases with the seniority.

Many PTAs are expected to do general testing on specific kind of products, which implies the importance of business domain knowledge. This can help testing the project more comprehensively. As they focus more on manual testing and exploring defects, testing skills, such as the ability to write test cases becomes quite important. As for automation test, it is ideal to have, but not as important as that of AT.

PT

Performance testing is the main aspect of non-functional testing to assure the quality or capability of a product and improve the user experience. The descriptions of responsibilities of PT were simply in the sample of job adverts. Figure 4.17 shows the extracted responsibilities of PT which were included in more than one advert. PT is mainly responsible for the non-functional aspects of the project.

PT is not discussed much in interviews. Main data came from job adverts. Initially, six job adverts for PT were collected. As more data were needed, seven job adverts were added. So, a total of 13 job adverts were collected. Figure 4.18 illustrates the

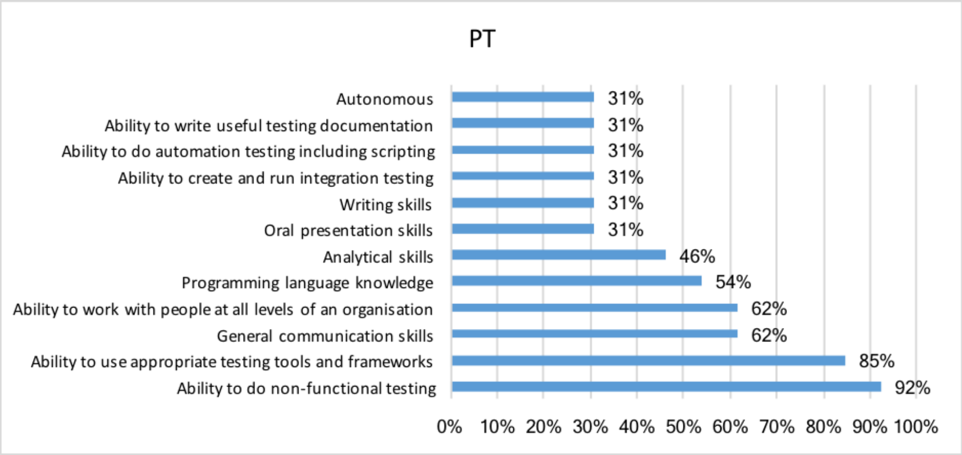


Figure 4.18: The Top Ten Skills and Capabilities for PT from Job Adverts

S&Cs required for PT.

As PT need tools to simulate the real scenarios of end users, usually the worst situations, "Ability to use appropriate testing tools and frameworks" is emphasised by employers. PT should also have "Programming language knowledge" to understand the technical design and code. This may help them find the bottleneck of performance.

"Ability to work with people at all levels of an organisation" was a special skill that was emphasised for PT in job adverts. The frequency reached as high as 62%. As non-functional requirements sometimes are not so clear as functional requirements, they have to interact much with different stakeholders, such as product owners and developers.

4.5 Skills and Capabilities for Seniority Levels

This section was designed to answer RQ3: What are the most important skills and capabilities of testers at different levels, and why?

Data came from three different sources: job adverts, interview questions Q14-16 and structured questions in Q17. The frequencies obtained from job adverts try to answer RQ3 from the perspective of recruiters. Q14-16 focused on the difference of the S&Cs between different seniority levels. Q17 was used to gather the reasons for the importance. Similar to the above two sections, the analysis was conducted on both skill categories and S&Cs. The frequency of the S&Cs in the job adverts was used to reflect employers' preferences. However, it is hard to extract the importance levels of the S&Cs from job adverts. Interviews questions were designed to deepen the understanding of the importance and reasons for the importance of them. The answers to those questions implicitly show the differences among the essential skills required by different seniority.

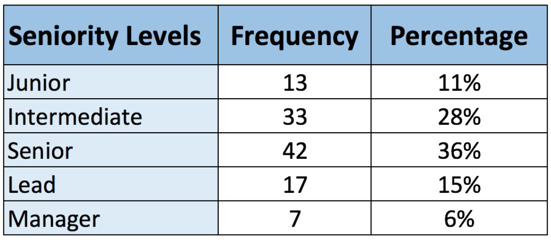
4.5.1 Different Seniority Levels for Testers

One of the preliminary findings is the identification of the seniority levels of testers. This was required to answer RQ3.

The names for different seniority levels were easily determined because they were conventional in job adverts. Five different seniority levels were identified: junior, intermediate, senior, lead and manager. The first five levels could be combined with those roles identified above. Take the lead for example, the job title, or whole role name for a tester could be "test lead", "automation test lead", "performance test lead", "QA lead", etc.. TM is defined as a generic test with top level in testing in this thesis. They are distinct from those specialists. TM refers to the resource person who is responsible for the development of the entire department, such as the strategies, policies and quality criteria. This role is also responsible for the personnel of the team, including recruiting, assessing and managing testers.

The frequencies of the levels spotted from job adverts are demonstrated in Table 4.21. It is clear that senior (36%) and intermediate (28%) were high in demand, while junior (4%) and manager (5%) was not so prevalent in the NZ market at the time of collecting the data. Besides, very few graduate tester position can be found in Seek and

Table 4.21: Frequency and Percentage of Different Seniority Levels Found in Job Adverts



Seniority Levels	Frequency	Percentage
Junior	13	11%
Intermediate	33	28%
Senior	42	36%
Lead	17	15%
Manager	7	6%

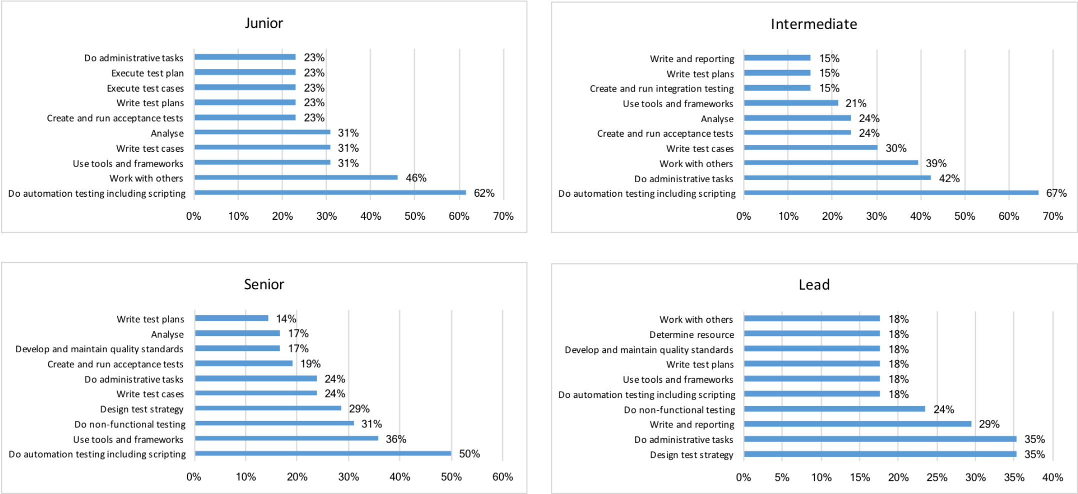


Figure 4.19: The Percentage of Responsibilities for Seniority

no graduate testers were found at the time collecting job adverts.

4.5.2 Responsibilities of Different Seniority

There are five different levels for testers: junior, intermediate, senior, lead and manager. They have different responsibilities and requirements.

Figure 4.19 shows the responsibilities obtained from job adverts. Due to the sample size of manager, further investigations are needed. The size of samples for junior and lead are both less than 20. The results for them can only be considered as some indications.

From the data of job adverts, the main responsibilities of junior testers were to learn automation testing and how to work with others. When they have progressed to intermediate levels, they began to do some administrative tasks. Senior testers had to decide what tools to use and write test frameworks for the project. Besides, they were responsible for the non-functional testing and can design test strategies when needed. Test lead was the person to develop and establish the test strategies and administer all the test activities, such as issues on product delivery and report progress to stakeholders.

The most apparent difference between different seniority levels obtained from interviews are elaborated as follows.

Juniors are usually expected to complete simple tasks assigned, such as writing or executing test cases and review specifications. They may be equipped with some fundamental technical knowledge or come from IT background, rather than having much business domain knowledge. That technical knowledge can help testers to understand the principles behind the code and design. Juniors may be told what to do rather than exploring on their own.

Compared with juniors, intermediate Testers are those who are capable of completing testing tasks in a project without intensive supervisions. They have already been familiar with the general process of testing and different techniques in testing. They may have been involved in various testing activities and have more experience and knowledge on both technical and business domains than juniors.

Senior testers have expertise in specific or multiple testing areas including manual and automated testing. Such roles are able to guide the direction of the whole project. One of their key responsibilities is to mentor and coach the juniors.

The major responsibilities of a test lead are to make plans and strategies, decide test tools and methodologies, review work flows and make sure the team works well. They are responsible for the tackling the issues of a simple project, which may involve project strategy, delivery, as well as the management and coordination of the whole team. This role takes more responsibility in administration than that of the senior testers.

However, test lead is not just the person who send out testing tasks, but who make the team grow and work well. P5 described a good test lead to the person who leads the team to a better situation. "He has the ability to grow the skills of his followers, rather than dictating them." Besides, based on the feedback from some interviewees, such as P3, test leads are those senior testers who intend to move towards to test managers. For example, P9 explained:" some roles and responsibilities of test managers, are taken by scrum masters in an agile team, such as communicating with different teams."

The differences between test lead and manager may include:

- Test manager focus on the whole development of the department and is responsible for the policy and quality criteria, while test lead may be just the lead of the team;
- Test manager may be responsible for multiple projects while test lead may just responsible for one project;
- Test manager may be just familiar with business domain, not necessary to know much technical knowledge;
- Test manager cares more about the project delivery, rather than the testing plans and techniques.

Table 4.22 summarised the difference between different seniority levels.

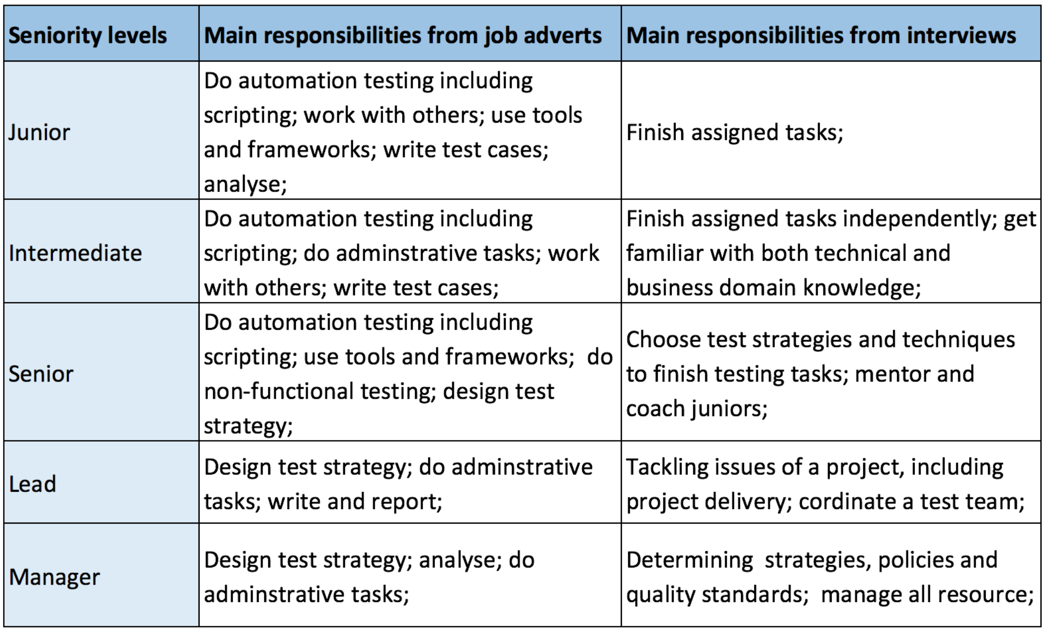
4.5.3 Skills and Capabilities for Different Seniority

Findings on Skill Categories

Figure 4.20 lists the weights of different skills mentioned in job adverts.

According to the data shown in the figure, managers had entirely different skill set with others. For test managers, communication skills and social skills were most

Table 4.22: Difference between Different Levels for Testers

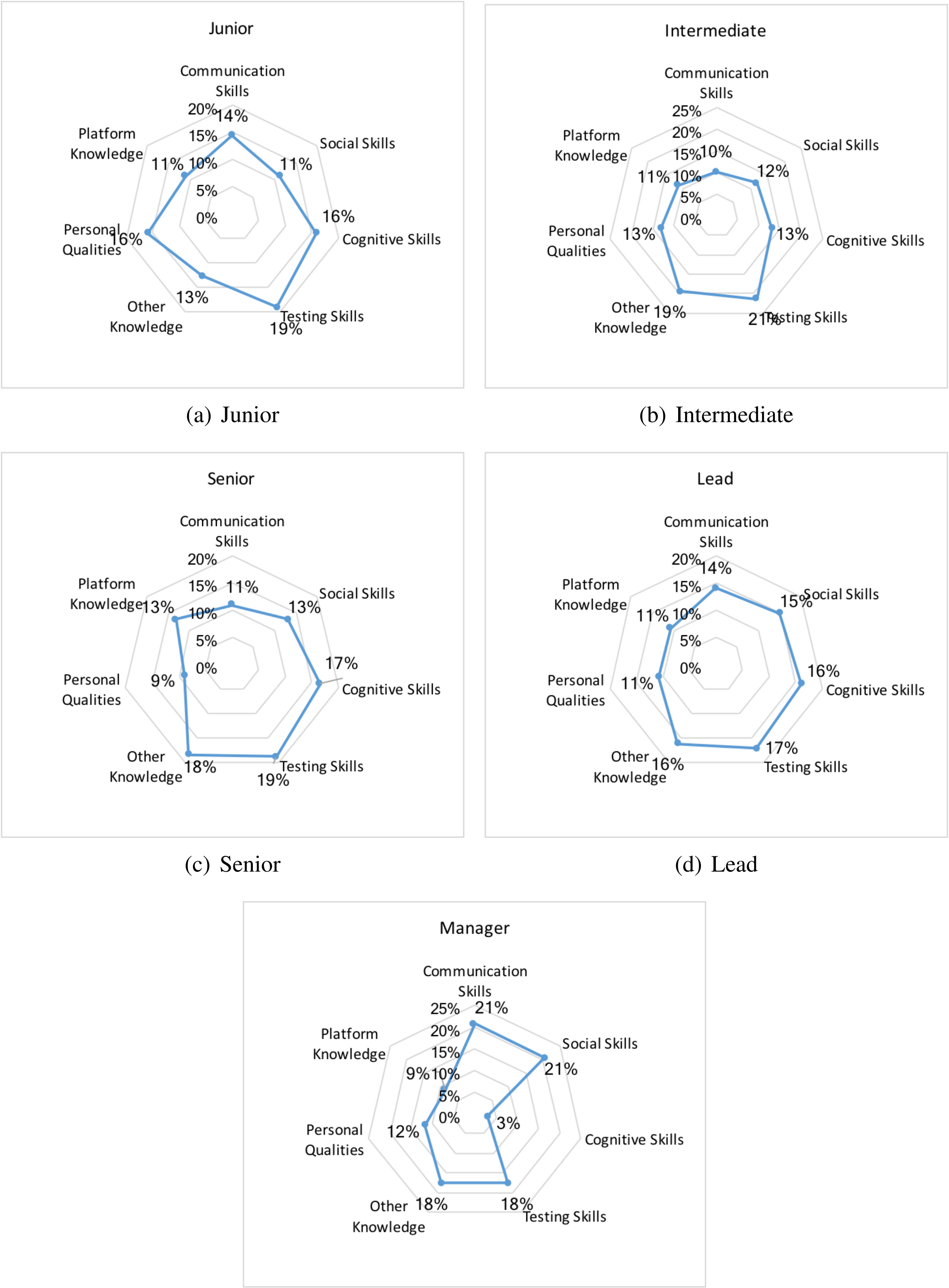


Seniority levels	Main responsibilities from job adverts	Main responsibilities from interviews
Junior	Do automation testing including scripting; work with others; use tools and frameworks; write test cases; analyse;	Finish assigned tasks;
Intermediate	Do automation testing including scripting; do adminstrative tasks; work with others; write test cases;	Finish assigned tasks independently; get familiar with both technical and business domain knowledge;
Senior	Do automation testing including scripting; use tools and frameworks; do non-functional testing; design test strategy;	Choose test strategies and techniques to finish testing tasks; mentor and coach juniors;
Lead	Design test strategy; do adminstrative tasks; write and report;	Tackling issues of a project, including project delivery; cordinate a test team;
Manager	Design test strategy; analyse; do adminstrative tasks;	Determining strategies, policies and quality standards; manage all resource;

frequently asked, while cognitive skills were ignored in many job adverts. The possible reason might be that they are assumed to be default abilities for managers. Managers needed high cognitive skills in many aspects, thereby making decisions on project issues and solving various problems for testers. One of main responsibility of TM was to manage the whole team and the delivery of projects rather than finish detailed testing tasks.

Testing skills were most frequently expected skills in four of those categories (except manager), which is consistent with that of all testers. However, the importance and requirements of testing skills were distinct for different seniority levels. It seems that the seniority was determined by the standards of the testing skills and other knowledge. For instance, junior testers were required to have a basic understanding on web domain knowledge, but senior testers were expected to possess a solid understanding on the area and solve related problems.

The skills required for test lead were relatively balanced. Test lead should both technical knowledge to conduct the project and soft skills to coordinate testers.



(a) Junior

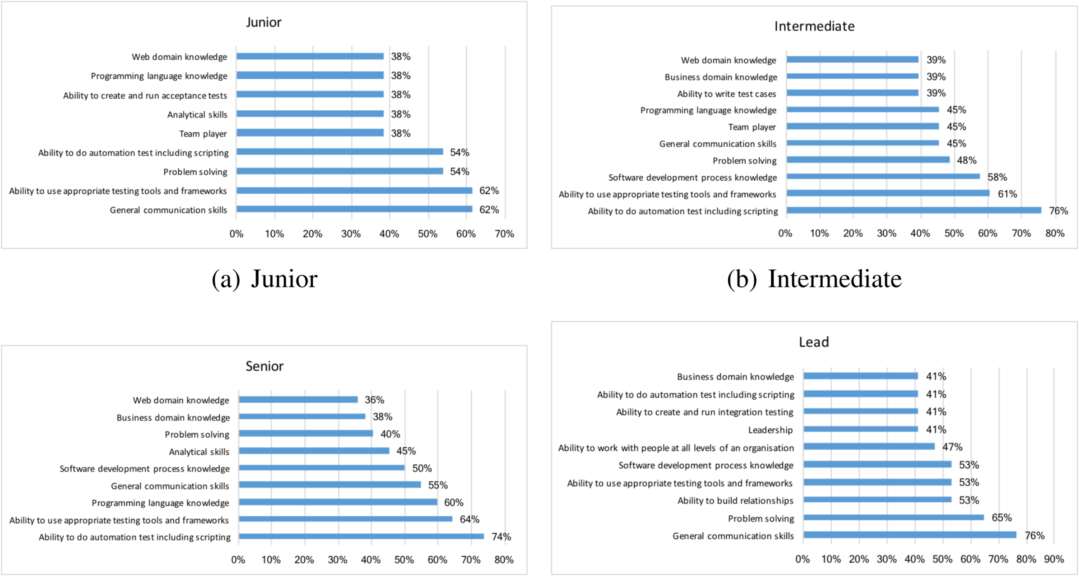
(b) Intermediate

(c) Senior

(d) Lead

(e) Manager

Figure 4.20: The Weights of High Level Skills for Different Seniority Levels



(a) Junior

(b) Intermediate

(c) Senior

(d) Lead

Figure 4.21: The Percentage of Top Skills for Different Seniority

Although the weights of personal qualities in Figure 4.20 various for 9% to 16%, the assigned importance levels indicated that the difference of personal qualities for testers was tiny for different seniority levels. For example, no matter what level of testers you were, you should be responsible.

Top Skills and Capabilities for Different Seniority

Findings obtained from job adverts

Figure 4.21 shows the top S&Cs for different seniority levels based on the percentage of a number of job adverts that mentioned related items. The rank of these skills can reflect the items that have caught recruiters' attention, although it cannot denote the importance of related S&Cs. There was just seven job adverts were found for TM, and the results of TM are not shown in the figure.

In the figure, the names of the top S&Cs remains basically the same, but the orders various. As mentioned above, "Ability to use appropriate tools and frameworks" was most frequently mentioned skills for software testing. For junior testers, general communication skills and problem solving skills were prominent, while for intermediate, "software development process knowledge" was significant. The importance of "Programming language knowledge" grew with the testing experience and seniority levels, until reaching the peak at senior testers. Then, for test Lead, "Problem solving" and "Ability to build relationships" were emphasised. The importance of "Team player" was gradually replaced by leadership when the role level increase.

Findings obtained from interview questions Q14-16

Interview questions Q14-16 were particularly designed to answer RQ3.

Q14. What are the most important differences in skills and capabilities between junior and intermediate testers?

Q15. What are the most important differences in skills and capabilities between intermediate and senior or lead testers?

Q16. What skills do you think you would need to develop to become a (a) test lead (b) test manager?

Answers to Q14-16 reflected the most impressive S&Cs that were recognised by interviewees and their opinions of interviewees about the difference on seniority levels.

For junior testers, they were expected to have the shared understanding that quality was the responsibility of the whole team and be able to work well others. They should have the ability to execute test cases and begin to learn how to design and write test cases. Technical domain knowledge was a bonus for them.

When testers shifted from junior to intermediate, their knowledge on programming languages and the ability to use the tools increased. They were expected to be more autonomous and formulate their own way of thinking. As an intermediate tester, one should begin to work independently and know the right way to finish assigned tasks.

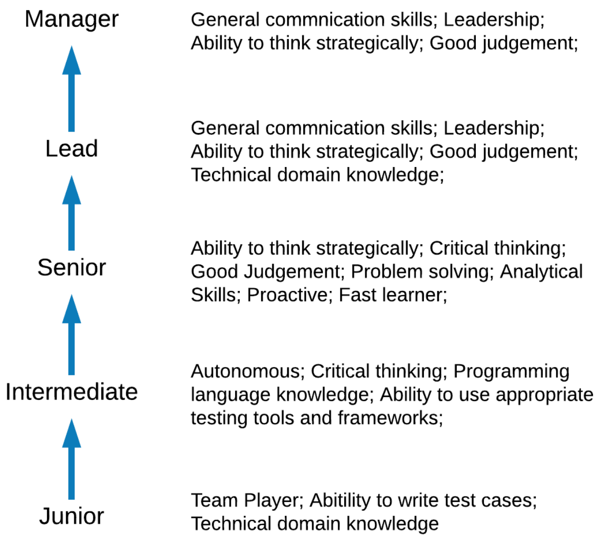

Manager
General commnication skills; Leadership; Ability to think strategically; Good judgement;
Lead
General commnication skills; Leadership; Ability to think strategically; Good judgement; Technical domain knowledge;
Senior
Ability to think strategically; Critical thinking; Good Judgement; Problem solving; Analytical Skills; Proactive; Fast learner;
Intermediate
Autonomous; Critical thinking; Programming language knowledge; Ability to use appropriate testing tools and frameworks;
Junior
Team Player; Abitility to write test cases; Technical domain knowledge

Figure 4.22: Top Skills and Capabilities for Different Seniority from Q14-16

This means one should complete test tasks by oneself, without much guidance and supervision. Cognitive skills they were sharpening, such as critical thinking and ability to think strategically, appear to be essential to the senior.

Senior testers had accumulated much experience in both testing skills and other knowledge compared to intermediate testers. They were expected to think more comprehensively and have an overall understanding on the whole project. Also, as a senior, one was expected to be able to learn new techniques proactively and had better skills in problem-solving and make a judgement. Besides, seniors had to think strategically rather than finishing assigned tasks.

TL was usually promoted from seniors and are expected to have all the abilities of seniors. The significant difference was leadership. They were supposed to manage a project and coordinate a test team. The lead was not only responsible for tackling the issues of a project, but also for the growth of the team members.

According to the responsibilities described above, the most important skills for TM were communication skills and leadership. Also, they had not just to think strategically, but also from the company's respective. Good judgement was needed in many situations and resources.

Explanation of the Importance of Skills and Capabilities

Although many of the skills mentioned have been explained in the former sections, their importance is briefly explained from the perspective of seniority levels.

Ability to create and run acceptance tests: Acceptance test is to think and test from the users' point of view (P9). It is an important testing activity affecting the delivery of the project. Senior testers need to do acceptance testing every day.

Ability to create and run integration testing: It is also daily work for the senior tester. However, the integration testing in an agile team is quite diverse. It can be done both manually and automatically.

Active listener: It is one of the most mentioned and emphasised skill, especially for junior testers. As junior testers may have not built their own views on different issues, they may have to be told what to do and have a checklist of their tasks. Listening actively helps them get familiar with what the developers have done, what test practices the team is using and how to finish tasks with a specific standard.

Problem solving: This is a baseline skill for the senior tester. They have to solve various problems testers may meet, even in an unfamiliar domain. When intermediate or junior testers encounter some issue out of their abilities, senior testers have to find a possible method.

Self-motivated: This helps testers to finish the tasks well, especially when facing some difficult situation. For example, when one has not found any bugs in a long period or when one got stuck in reproduce the bugs, self-motivation may encourage one to go on.

Curious: As one of tester's responsibility is to discover defects in artefacts (from P5), curiosity may conduct testers to find more bugs and causes for them.

Table 4.23: Difference between Top Items in Skills Categories for Seniority

Skill Names	Junior/Intermediate Testers	Senior Testers
Testing skills	Ability to do non-functional testing; Ability to write test cases;	Ability to create and run integration testing; Ability to create and run acceptance tests;
Other knowledge	Software development process knowledge;	Business domain knowledge;
Personal qualities	Attention to details; Curious;	Willing to learn/share; Attention to details;

Good judgement: It is very important for senior testers. They have to evaluate the risks of what developers have done and make a lot of decisions on what to test, when and how to test, as well as where to put effort. Think of this scenario: At the end of a sprint, you find some tasks cannot be finished. Then you have to decide whether you want to move it to the next sprint or not. Besides, they have to review or judge the work of junior and intermediate testers.

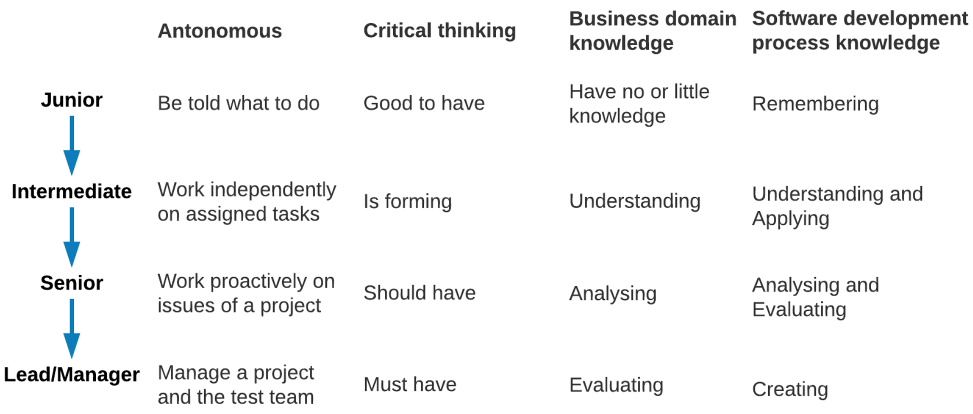
To get more detail about the most important S&Cs in each skill category, data are presented in the figures in Appendix F. The top skills in each category are highlighted. The most significant difference between the results are skills in testing skills, other knowledge and personal qualities.

Importance Levels of Skills and Capabilities

According to the findings from the interviews, it is worth mentioning that although the top skills may be the same for different seniority levels, their expectation of skill levels varies. Some items, such as "attention to details" and "willing to learn/share", are irrelevant with seniority levels. But many items, especially those in "testing skills" and "other knowledge", are closely associated with seniority levels.

Figure 4.23 are some examples how the importance levels change with seniority levels.

Take Business Domain Knowledge (BDK) as an example. As junior testers, they



	Antonomous	Critical thinking	Business domain knowledge	Software development process knowledge
Junior	Be told what to do	Good to have	Have no or little knowledge	Remembering
Intermediate	Work independently on assigned tasks	Is forming	Understanding	Understanding and Applying
Senior	Work proactively on issues of a project	Should have	Analysing	Analysing and Evaluating
Lead/Manager	Manage a project and the test team	Must have	Evaluating	Creating

Figure 4.23: Important levels of Some Samples

actually are not expected to have much depth of understanding of business domain knowledge; some basic testing knowledge from ISTQB tutorial is enough. When becoming an intermediate tester, he or she is expected to have accumulated rich BDK to finish testing tasks independently. BDK can help them to understand the real requirements and find more defects in different artefacts. Then, as a senior test, one should have his own thinking about the requirements and discuss with related stakeholders, such as product owner.

This is similar to other knowledge, such as "Software development process knowledge" and "Ability to use appropriate testing tools and frameworks". Usually, it is the senior testers' responsibility to choose them, while intermediate testers just use them. But juniors may be fresh to the tools.

Actually, the levels of testing skills and knowledge can be identified using the Anderson and Krathwohl's Taxonomy. They can thereby be applied in a testing course.

4.6 Impact of Findings for Stakeholder Categories

This research has several objectives to benefit different groups of people: employers, educators, testers, and other researchers. This section elaborated the possible ways to use the findings and achieve those objectives. They are based on information on generic S&Cs needed for any tester, as well as those for different testing roles and seniority levels.

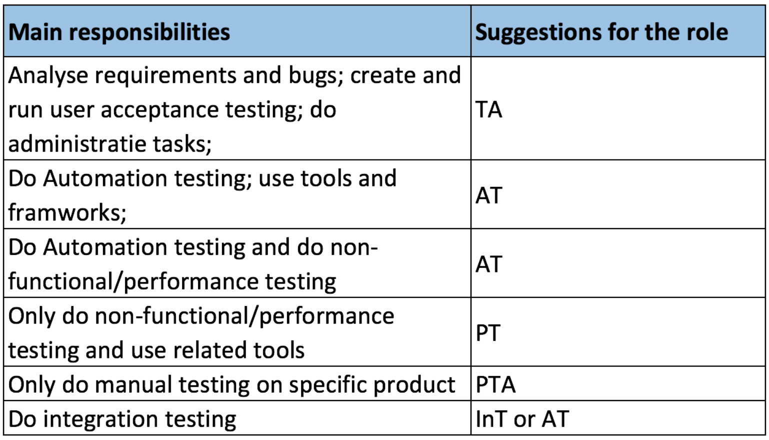
To Employers

Data from the sample of job adverts provided the expectation of advert writers, who may be a TM, someone from HR department or an employment agency. A job advert should consider the real requirements of the vacancy and expectation of the employer, no matter who wrote it. Employers who want to recruit a tester can design or refine their job adverts based on the knowledge gained in the research.

This research has revealed both the most favourable S&Cs that recruiters think they want and those most valuable ones that test professionals need in a practical context. According to the research conducted by a software consultant (Tockey, 2015), survey on job postings is an effective way to identify the gap between what employers ask for and what they actually need and the necessary to align recruiting criteria with industry requirements. There was some difference between what was asked in adverts and what test practitioners said they needed. So that could mean the adverts writers were out of touch with the reality of testing and therefore there was a risk that would get people with the wrong skills. So we suggest job adverts talk more closely with the testers in the team to design the job adverts, to involve the skills that are truly reflecting what they need to do the job level and role they are looking after.

Based on the information provided in this research, they can quickly work out what role and seniority level they want and identify the S&Cs that they prefer to involve in the job advert. Findings from both job adverts and interviews can be utilised when an

Table 4.24: Suggestions for Identification of a Testing Role



Main responsibilities	Suggestions for the role
Analyse requirements and bugs; create and run user acceptance testing; do administratie tasks;	TA
Do Automation testing; use tools and framworks;	AT
Do Automation testing and do non-functional/performance testing	AT
Only do non-functional/performance testing and use related tools	PT
Only do manual testing on specific product	PTA
Do integration testing	InT or AT

employer wish to recruit a tester.

To design or refine a job advert, possible steps are:

- Check difference in responsibilities and most important S&Cs between different roles and seniority levels and identify the role and seniority level they want to recruit. Employers may not have a clear understanding of the testing role and seniority level. For example, employers may not know if an intermediate or senior tester is needed for the vacancy. Their expectations on the candidates can be adjusted after reading the main findings on this research.

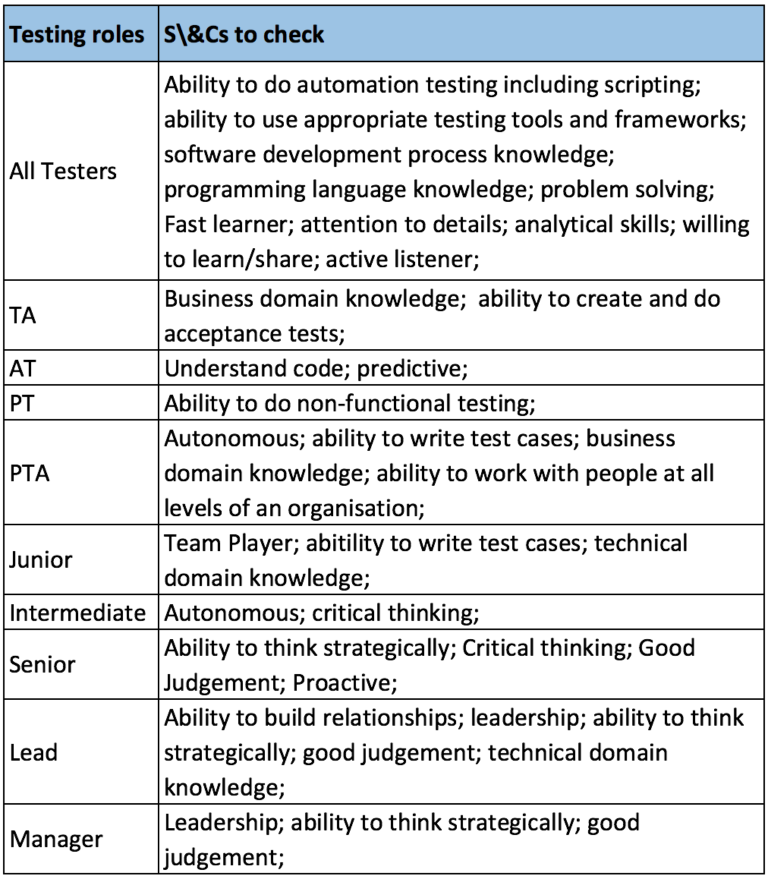
- Refine the S&Cs by checking the most important S&Cs for any tester and the reasons for the importance of those S&Cs.

- Check the particular S&Cs of the role and seniority level and make final decisions.

Table 4.24 and Table 4.22 shows the suggestions for identifying the role and seniority level respectively. In this table, the S&Cs for different role and seniority levels are ignored when they already existed in the row of "All testers".

Table 4.25 shows the main S&Cs to be checked by employers for the refinement of job adverts. S&Cs for all roles can be checked before roles and seniority levels.

Table 4.25: Skills and Capabilities to be Checked for Each Role



Testing roles	S\&Cs to check
All Testers	Ability to do automation testing including scripting; ability to use appropriate testing tools and frameworks; software development process knowledge; programming language knowledge; problem solving; Fast learner; attention to details; analytical skills; willing to learn/share; active listener;
TA	Business domain knowledge; ability to create and do acceptance tests;
AT	Understand code; predictive;
PT	Ability to do non-functional testing;
PTA	Autonomous; ability to write test cases; business domain knowledge; ability to work with people at all levels of an organisation;
Junior	Team Player; abitility to write test cases; technical domain knowledge;
Intermediate	Autonomous; critical thinking;
Senior	Ability to think strategically; Critical thinking; Good Judgement; Proactive;
Lead	Ability to build relationships; leadership; ability to think strategically; good judgement; technical domain knowledge;
Manager	Leadership; ability to think strategically; good judgement;

To Testers

To testers, information on the kinds of S&Cs that needed for a tester is provided, as well as those required for different roles and seniority levels. They can refer to those S&Cs for their own purpose. The findings and analysis in this research may benefit tester in the following ways:

- To help test practitioners benchmark themselves by checking the S&Cs needed in their current role and making some improvements correspondingly. They may focus on the S&Cs needed for their current role and those for higher seniority levels. For a junior AT, one can check the S&Cs for AT, as well as for junior and intermediate testers; for an intermediate AT, one can focus on those for AT, as well as for intermediate and senior testers. Other roles and seniorities can use the findings similarly.

- To help those testers or "future" ones seeking a testing job to discover the skill-set they already have and get prepared for a specific role in the testing market. This is especially useful to those who are not a tester yet. They can find the most important S&Cs that they have already obtained and highlight them in their resume and interviews, or guide them to strengthen their weak ones. This is also helpful to test practitioners who may want to shift from one role to another, such as from TA to AT.
- To help test professionals recognise some important skills ignored by themselves and gain a comprehensive understanding of the topic. From the interviews, we found the S&Cs highlighted by test practitioners showed diversity. When they were asked to fill the forms about the importance levels, they reached more agreements on several items. This research adds knowledge about S&Cs customised for testers, which can be used as a checklist.

Table 4.25 can be employed as the main checklist while reasons for their importance elaborated in the analysis sections can be used as a reference or explanation. For example, a junior AT, S&Cs for "All testers", "AT", "Junior" and "Intermediate" could be considered. Similarly, for an intermediate TA, S&Cs for "All testers", "TA", "Intermediate" and "Senior" may be checked.

To Educators

Understanding the generic S&Cs to all testers, as well as those to specific roles and seniority levels can inform the design of the course to develop these skills, in conjunction with their expertise in pedagogies.

This research reflects the most important S&Cs by employers and test professionals. Those items were preliminarily validated by industry. It may be used as an indication on the "real" requirements of the market. Students can benefit from this practical course and become a qualified tester after formal and systematic training.

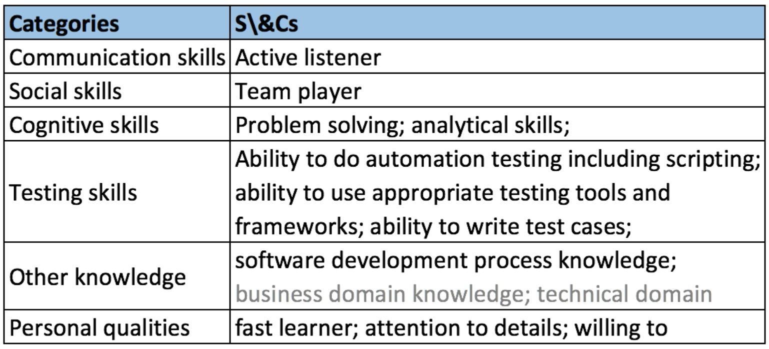
From the results of LR, most existing courses in testing focus on the practical experience on real projects. The most common skills, such as analytical skills, problem solving skills and teamwork, are targeted. The most famous skill model used in testing courses, namely CDIO, is initially designed for common software engineering education. But tester is a particular role in software engineering and has its own requirements on S&Cs. This is why the results of this research work.

To make the course more practical and customised, educators may first identify the objectives of the course and make decisions on what role and seniority level of testers they want to cultivate. The information on the S&Cs related to this role and seniority level can be considered.

For educators, if they want to design a course that aims to design for junior testers, we assure they want to get a TA job because TA is the most common job found in testing market in NZ. S&Cs for junior TA may include: ability to do automation testing including scripting, active listener, problem solving, ability to use appropriate testing tools and frameworks, business domain knowledge, software development process knowledge, fast learner, attention to details, analytical skills, willing to learn/share, team player, ability to write test cases. Table 4.26 shows the S&Cs in different category. Business and technical domain knowledge are not suggested to include in a testing course because they are too broad and is difficult to be taught in a testing course with limited classes.

Educators may also be interested in an AT role; there is some difference from that course for a TA. When designing a course for AT, two other S&Cs should be included: understanding code and predictive. The most popular testing tools and language identified in this research, namely selenium and java, could be used for practice part of the course. Students should be taught how to evaluate the influence of

Table 4.26: Skills and Capabilities for Junior TA when Designing a Course Targeted



Categories	S\&Cs
Communication skills	Active listener
Social skills	Team player
Cognitive skills	Problem solving; analytical skills;
Testing skills	Ability to do automation testing including scripting; ability to use appropriate testing tools and frameworks; ability to write test cases;
Other knowledge	software development process knowledge; business domain knowledge; technical domain
Personal qualities	fast learner; attention to details; willing to

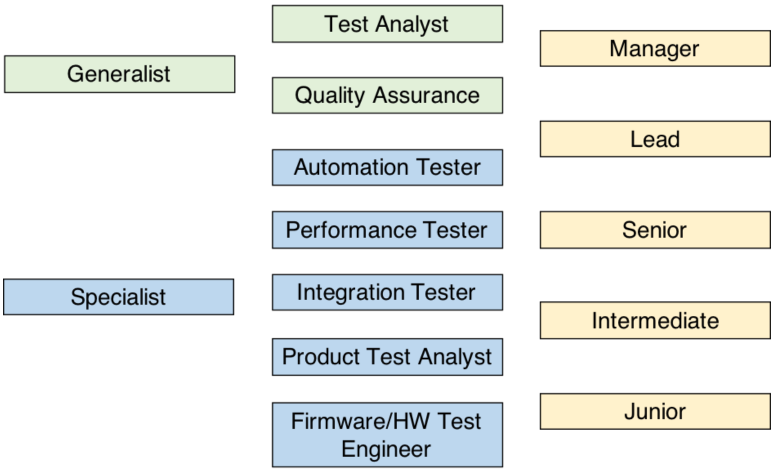


Figure 4.24: Structure of Role and Seniority Identified in this Research

what developers have done and then decide where to apply test automation.

To Researchers

For researchers, a conceptual model for the S&Cs provides a structure for future research. It includes the list they can talk about in the S&Cs of testers and their meanings. They are presented in section 4.2.

This research has also provided some structure in the roles and seniority levels that could be reused by other researchers (see Figure 4.24).

4.7 Conclusion

As no suitable framework has been found for testers in LR, a customised conceptual model was developed by analysing the job adverts. It was revised based on the findings in the LR and interviews. The framework is presented in Section 4.2. It is a two-hierarchical conceptual model, in which 60 items are classified into seven different categories.

Section 4.3 answered the generic S&Cs for any tester. The most desirable skills and capabilities that employers expect were collected from job adverts, while the most valuable ones for test professionals were identified from interviews performed in this research. Data were analysed on both levels of the proposed model and findings were triangulated from both perspectives.

Section 4.4 was designed to answer RQ2: the most important S&Cs for each testing role. Six different roles were identified and used to group the findings of each role.

Similar to Section 4.4, Section 4.5 answered RQ3. Five seniority levels, namely junior, intermediate, senior, lead and manager, were identified through their different responsibilities. They were then used to organise the findings on seniority levels. A general understanding on the differences of S&Cs for each seniority level was discussed.

After answering the RQs, the impact of those findings for different stakeholders was discussed and potential usage of the findings was elaborated.

Chapter 5

Conclusion

5.1 Introduction

This research aims to gain an understanding of the most important S&Cs needed by contemporary software testers for different roles and seniority levels. The findings of this research should be helpful for recruiters to design or refine job adverts, for testers to guide their professional development, for course designers to cultivate testers based on appropriate S&Cs, as well as for researchers to conduct similar research.

A two-pronged approach was taken to seek answers to the RQs. Firstly, online job adverts for testers were collected and analysed. A customised conceptual model of S&Cs for testers was generated based on the overall list of S&Cs extracted from job adverts. Then, the generic S&Cs for any tester, as well as those most important for different testing roles and levels of seniority were identified using frequency of them in the sample of adverts. The second part of the research was conducted to answer the RQs from the perspective of test professional. Interview questions were designed based on the RQs and the initial model of S&Cs mainly came from job adverts. Semi-structured questions were mainly designed to discover most valuable S&Cs to testers from the perspective of test practitioners and the context of the interviewees and the organisations they were working. On the other hand, structured questions were used to collect their view on the levels of importance on each S&C using Likert scale. Interviewees were asked to speak out loud and an understanding of why they marked each S&C as more, or less, important was obtained. The findings from different sources were triangulated to deepen the understanding.

To achieve the research aims and objectives, particular analysis approaches were select for different source data. The process analysing data from job adverts was an HC. Content analysis was performed to extract and code the S&Cs needed for testers in job adverts and quantitative analysis on frequencies was used to identify the importance of each S&C. In addition, thematic analysis was conducted for data from semi-structured interview questions and distribution of frequency and median were employed to analyse the importance levels of S&Cs in the proposed model. The reasons for the levels of importance collected from speak-out loud protocol were grouped by S&Cs to strength the findings.

This chapter is organised as follows: Section 5.2 answers the RQs and summarises the contributions of this research. Section 5.3 presents the limitation and possible future work.

5.2 Main Findings

This section firstly elaborates the answers to the three RQs and then the utility for different stakeholder groups.

5.2.1 Answers to the Research Questions

All the questions are answered from the perspectives of both employers and test professionals. The three most important S&Cs are listed in the subsection.

RQ1: What are the most important generic skills and capabilities needed by any tester, and why?

According to the data of job adverts and literature review, employers were more likely to underline testing skills and other knowledge, while test professionals emphasised more on personal qualities and cognitive skills.

The most favourable S&Cs that employers wanted included ability to do automation test including scripting, ability to use appropriate testing tools and frameworks, general communication skills, and so on. Test automation was a very important method to do testing in modern software development environment because testers can get involved in the development process to the earliest stage.

However, S&Cs, such as attention to details, willing to learn/share and fast learner were highlighted by interviewees. Attention to details is the most important skill generic to any tester. It helps testers to discover defects in different artefacts during the development process and locate causes effectively and efficiently. Willing to learn/share and fast learner is becoming important because of the fast-updating techniques. Willing to learn/share can help testers get a broad view on various new technologies and methodologies and grow fast to a good tester, while fast learner may help testers keep pace with the changing practices in the knowledge they need to know.

Besides, the top three S&Cs found in LR are domain knowledge, general communication skills and teamwork.

RQ2: What are the most important skills and capabilities that are specific for different testing roles, and why?

Six roles, namely TA, AT, QA, PT, PTA and TE were obtained after content analysis. TA and AT were representative roles of generalist and specialist and the most important S&Cs were compared and discussed. As only four and two job adverts of QA and TE were found during the period of collecting data, the most important S&Cs of them were not discussed.

The top three S&Cs for AT identified in job adverts were the ability to do automation test including scripting, ability use appropriate testing tools and frameworks, as well as programming language knowledge. This is because the main responsibilities of AT are to do automation testing using different tools or programming languages.

Those for TA were the ability to do automation test including scripting, general communication skills and problem-solving. Although some TAs were also required to do automation testing, the percentage of this skill needed for AT was much higher. The difference between AT and TA were determined by their responsibilities. TA is responsible for acceptance test and frequent communication with different stakeholders are needed. Besides, they may meet various problems at different levels that need to be solved by themselves before convincing related stakeholders.

The difference between AT and TA were: TA focused more on getting a broad view in technical domain knowledge while AT was asked to understand the code and make decisions on where to do automation testing.

The identified most important S&Cs for PTA were autonomous, general communication skills and problem solving. PTA is a special TA who focus on the specific product or business domain. PTA is expected to be very familiar with business domain knowledge and can work independently. The reasons for the importance are similar with TA.

The top skills and capabilities include: ability to do non-functional testing, ability to use appropriate testing tools and frameworks, general communication skills. PT is a specialist on non-functional testing. "ability to use appropriate testing tools and frameworks" and "programming language knowledge" is especially important to them.

RQ3: What are the most important skills and capabilities of testers in different levels, and why?

In this thesis, five levels, namely junior, intermediate, senior, lead and manager, were determined and analysed.

From job adverts, top S&Cs for junior testers were: general communication skills, ability to use appropriate testing tools and frameworks, as well as problem solving. Those for intermediate were: the ability to do automation test including scripting, the ability to use appropriate testing tools and frameworks, as well as software development process knowledge. To seniors, they include the ability to do automation test including scripting, ability to use appropriate testing tools and frameworks, as well as programming language knowledge. Those for test lead were general communication skills, problem solving and ability to build relationships. General communication skills, leadership, business domain knowledge were identified as most important to test managers. In the data extracted from job adverts, TM showed the dominant difference from other levels. Communication skills and social skills were highlighted for TM, while cognitive skills were neglected.

From the data of interviews, the top S&Cs are listed as follows. Junior: team player, active listener and technical domain knowledge. Intermediate: autonomous, critical thinking and programming language knowledge. Senior: critical thinking, good judgement, and problem solving; Lead: general communication skills, ability to think strategically and technical domain knowledge. Manager: leadership, general communication skills, and ability to think strategically.

For juniors, their most important responsibilities were learning to finish regular testing tasks, such as executing test cases and reporting bugs under guide, as well as get a shared understanding with the team. As an intermediate, they were expected to finish tasks independently and begin to think critically to get prepared for seniors. Knowledge of programming languages was expected to understand developers' language and communicate better with them. For seniors, high cognitive skills, such as critical thinking, good judgement and problem-solving skills, were unavoidable because they were expected to be able to handle issues in all testing activities independently. Skills for lead and manager were obvious.

5.2.2 Contributions

The main contributions of this research are adding knowledge to the S&Cs for testers. It provides potential readers insights into the most important S&Cs for different testing roles and levels of seniority and the reasons for the importance. Those findings may benefit different stakeholders in the following ways.

- They can help employers design or refine job ads and recruit more appropriate testers. Employers can get a clear understanding of what role and level of testers they need and check the list of the most important S&Cs for their vacancy.

- They can help tester practitioners benchmark themselves by checking the S&Cs for the role they want to be and make improvements correspondingly. It can also help testers who are seeking a testing job to highlight the S&Cs in the list that they have already obtained.

- They can help educators design practical tertiary-level course in testing. Many test courses used the skills extracted from frameworks for software engineering education or without not much evidence for the skills they chose. In this thesis, suggestions were given for those who want to design a course for different roles and seniority levels, such as junior TAs or ATs.

- They may also contribute to researchers. The outputs of this thesis include a structure of testing roles and seniority levels, as well as a customised conceptual model for the important S&Cs for software testers, which can be reused by other researchers.

5.3 Limitation and Future Work

In this research, two different sources of data were used to seek answers to RQs from perspectives of both recruiters and test professionals. The procedural details were provided in Chapter 3 to minimising threats to internal validity. Knowledge was added on the S&Cs generic to any testers, as well as different roles and seniority levels. Some useful and interesting patterns were found. However, limitations exist and future research can be conducted.

The results of the study may not be generalisable to other countries, where the job market and way of working may differ, since this study is only based on NZ. Some differences may exist in other countries or area because the S&Cs for testers are influenced by different factors, including the country's culture, values, ways of working and stage of development.

Also, bias may exist when interviewees answering the questions. Their performance might be affected by various factors, such as how the interviewer prompted the semi-structure interview. In order to make sure the interviewees had the same understanding of the questions and vocabulary as the researchers, the terminology and aims were explained prior to the interviews and participants were invited to ask clarification questions at any stage of the interview. The presence of the researchers may have influenced the interviewees. To help establish trust we had an informal conversation between researchers and interviewees prior to the interview.

To avoid ambiguity and leading questions, a pilot of the interview questions was undertaken and the questions were reviewed by two researchers. The same interview protocol and interview guide were used consistently throughout all interviews.

Another bias may occur with the selection of interviewees. Our participants were voluntary and were from a variety of organisations, seniority levels and job roles, reducing selection bias.

The likelihood of researcher bias in coding and categorising for thematic analysis was reduced by two researchers coding and categorising and discussing any disagreements until consensus was reached.

Also, as the S&Cs and their synonyms were extracted and grouped manually, certain items with very low frequency may be ignored. But the impact of this bias could be very slight because the top ten important S&Cs were chosen and the research focus more on why specific S&Cs are more or less important. In addition, the bias was reduced by using the accumulated frequencies and quantitative analysis.

Possible methods for further research are listed as follows:

- Further research could be done by analysis on job adverts. The data in this research is one snapshot of skills in job adverts. This could be extended to more samples over time and any dynamics and changes identified. In this thesis, quantitative data, or the frequency of skills and capabilities, was employed to analyse the importance of them. However, frequency was just one aspect of the importance. The level of importance should also be considered. In the collected job samples, there was no consistent way to describe the knowledge, skills and capabilities needed for the vacancies, which increased the difficulty to define and extract the importance levels from text description. Some job profiles used keywords, such as "essential", "necessary" and "bonus", to express the importance level, but not much. It would be good if more job adverts could be found for future research.

- Other research approaches, such as questionnaires could be considered to complement and triangulate the finding from structured questions of interviews or more interviews could be performed. In the interviews, grid value 1-5 were employed to explore the skill and capabilities levels. To research the importance levels of skills and capabilities, further research can be conducted using questionnaires to collect enough data for quantitative analysis and then be validated from the industry.

- From the findings described above, the skills and capabilities highlighted in job adverts and interviews were significantly different. Future research can be conducted to add weight to how convincing they are.
- The opinion of the stakeholders, such as developer, product owner, and business analyst, can also be considered to provide different perspectives because they are the members that testers work with.
- Further research can be performed to analyse how "agile" way significantly changes software testing, as well as the skills and capabilities needed for agile testers. . Agile is a prevalence methodology for software development. Agility has greatly improved the software industry, including the software testing. In this research, about half of the job adverts declared their context for software development as agile, while all the interviewees are using or trying using different agile methods. So, the findings also reflect those for agile testers to some extent. It could be extended to gain more in-depth understanding on the topic.

References

Acuna, S. T., Juristo, N. & Moreno, A. M. (2006). Emphasizing human capabilities in software development. *IEEE software*, *23*(2), 94–101.

Ahmed, F., Capretz, L. F., Bouktif, S. & Campbell, P. (2012). Soft skills requirements in software development jobs: a cross-cultural empirical study. *Journal of Systems and Information Technology*, *14*(1), 58–81. doi: 10.1108/13287261211221137

Ahmed, F., Capretz, L. F. & Campbell, P. (2012). *Evaluating the Demand for Soft Skills in Software Development* (Vol. 14) (No. 1). doi: 10.1109/MITP.2012.7

Ahmed, F., Capretz, L. L. F., Bouktif, S. & Campbell, P. (2013). Soft Skills and Software Development: A Reflection from the Software Industry. *Journal of Systems and Information Technology*, *4*(May), 171–191. doi: 10.4156/ijipm.vol14.issue3.17

Alexander, P. A. & Judy, J. E. (1988). The interaction of domain-specific and strategic knowledge in academic performance. *Review of Educational research*, *58*(4), 375–404.

Allen, I. E. & Seaman, C. A. (2007). Likert scales and data analyses. *Quality progress*, *40*(7), 64.

Almi, N. E. A. M., Rahman, N. A., Purusothaman, D. & Sulaiman, S. (2011). Software engineering education: The gap between industry's requirements and graduates' readiness. *ISCI 2011 - 2011 IEEE Symposium on Computers and Informatics*, 542–547. doi: 10.1109/ISCI.2011.5958974

Anderson, L. W. & Bloom, T. o. e. o., Benjamin Samuel. (2001). *A taxonomy for learning, teaching, and assessing : a revision of bloom's taxonomy of educational objectives.* New York : Longman, [2001].

Armour, P. G. (2001). The laws of software process. *Commun. ACM*, *44*(1), 1517.

Armour, P. G. (2005). The unconscious art of software testing. *Communications of the ACM*, *48*(1), 15–18. doi: 10.1145/1039539.1039554

Astigarraga, T., Dow, E. M., Lara, C., Prewitt, R. & Ward, M. R. (2010). *The Emerging Role of Software Testing in Curricula.* doi: 10.1109/TEE.2010.5508833

Attewell, P. (1990). What is skill? *Work and occupations*, *17*(4), 422–448.

Bajaj, S. K. & Balram, S. (2009). Incorporating Software Testing as a Discipline in Curriculum of Computing Courses. In J. Yang, A. Ginige, H. C. Mayr & R.-D. Kutsche (Eds.), *Information systems: Modeling, development, and integration: Third international united information systems conference, uniscon 2009, sydney, australia, april 21-24, 2009. proceedings* (pp. 404–410). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-01112-2_41

Beecham, S., Baddoo, N., Hall, T., Robinson, H. & Sharp, H. (2008). Motivation in software engineering: A systematic literature review. *Information and software technology*, *50*(9), 860–878.

Beer, A. & Ramler, R. (2008). The role of experience in software testing practice. *EUROMICRO 2008 - Proceedings of the 34th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2008*, 258–265. doi: 10.1109/SEAA.2008.28

Bertolino, A. (2007). Software Testing Research : Achievements , Challenges , Dreams. *Future of Software Engineering. FOSE '07*(September), 85–103. doi: 10.1109/FOSE.2007.25

Bin, Z. & Shiming, Z. (2013). Curriculum reform and practice of software testing. *International Conference on Education Technology and Information System (ICETIS 2013)*(Icetis), 841–844.

Bin, Z. & Shiming, Z. (2014, Aug). Experiment teaching reform for software testing course based on cdio. In *Computer science & education (iccse), 2014 9th international conference on* (pp. 488–491). doi: 10.1109/ICCSE.2014.6926509

Boell, S. K. & Cecez-Kecmanovic, D. (2010). Literature reviews and the hermeneutic circle. *Australian Academic & Research Libraries*, *41*(2), 129-144. doi: 10.1080/00048623.2010.10721450

Braun, V. & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77-101. doi: 10.1191/1478088706qp063oa

Briand, L. C. & Labiche, Y. (2004). Empirical Studies of Software Testing Techniques : Challenges , Practical Strategies , and Future Research. *WERST Proceedings/ACM SIGSOFT*, *29*(5), 1–3. doi: 10.1145/1022494.1022541

Buchan, J., Ekadharmawan, C. H. & MacDonell, S. G. (2009). Insights into domain knowledge sharing in software development practice in smes. In *Software engineering conference, 2009. apsec'09. asia-pacific* (pp. 93–100). doi: 10.1109/APSEC.2009.47

Causevic, A., Sundmark, D. & Punnekkat, S. (2010). An industrial survey on contemporary aspects of software testing. *ICST 2010 - 3rd International Conference on Software Testing, Verification and Validation*, 393–401. doi: 10.1109/ICST.2010.52

Cerqueira, P. (2016). Lessons Learned as a Tester in an Agile Scrum DW / BI Environment. *Business Intelligence Journal*, *21*(2), 10–17.

Clarke, V. & Braun, V. (2014). Thematic Analysis. In T. Teo (Ed.), *Encyclopedia of critical psychology* (pp. 1947–1952). New York, NY: Springer New York. doi: 10.1007/978-1-4614-5583-7_311

Clason, D. L. & Dormody, T. J. (1994). Analyzing data measured by individual likert-type items. *Journal of agricultural education*, *35*, 4.

Collins, E. F. & de Lucena, V. F. (2012). Software Test Automation practices in agile development environment: An industry experience report. *2012 7th International Workshop on Automation of Software Test (AST)*, 57–63. doi: 10.1109/IWAST.2012.6228991

Crossman, E. (1964). Information processes in human skill. *British medical bulletin*, *20*(1), 32–37.

Day, P. (2014). N-Tiered test automation architecture for Agile software systems. *Procedia Computer Science*, *28*, 332–339. doi: 10.1016/j.procs.2014.03.041

De Souza, É. F., Falbo, R. D. A. & Vijaykumar, N. L. (2015). Knowledge management initiatives in software testing: A mapping study. *Information and Software Technology*, *57*(1), 378–391. doi: 10.1016/j.infsof.2014.05.016

Deak, A. (2014a). A Comparative Study of Testers' Motivation in Traditional and Agile Software Development. *Product-Focused Software Process Improvement*, 1–16. doi: 10.1007/978-3-319-13835-0_1

Deak, A. (2014b). What Characterizes a Good Software Tester? - A Survey in Four Norwegian Companies. *Testing Software and Systems - Proceedings of the 26th IFIP WG 6.1 International Conference, ICTSS 2014, Madrid, Spain, September 23-25, 2014*, *8763*, 161–172. doi: 10.1007/978-3-662-44857-1

Deak, A. & Stalhane, T. (2013). Organization of testing activities in Norwegian software companies. *Proceedings - IEEE 6th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2013*, 102–107. doi: 10.1109/ICSTW.2013.18

Deak, A., Stålhane, T. & Sindre, G. (2016). Challenges and strategies for motivating software testing personnel. *Information and Software Technology*, *73*, 1–15. doi: 10.1016/j.infsof.2016.01.002

De La Harpe, B., Radloff, A. & Wyber, J. (2000). Quality and generic (professional) skills. *Quality in Higher Education*, *6*(3), 231–243.

Descombe, M. (2003). The good research guide. *For Small-scale Research Projects*.

Easterbrook, S., Singer, J., Storey, M.-A. & Damian, D. (2008). Selecting Empirical Methods for Software Engineering Research. *Guide to Advanced Empirical Software Engineering*, 285–311. doi: 10.1007/978-1-84800-044-5_11

Fernández-Sanz, L., Villalba, M. T., Hilera, J. R. & Lacuesta, R. (2009). Factors with negative influence on software testing practice in spain: A survey. In *European conference on software process improvement* (pp. 1–12).

Fleming, V., Gaidys, U. & Robb, Y. (n.d.). Hermeneutic research in nursing: developing a gadamerian-based research method. *Nursing Inquiry*, *10*(2).

Fu, L. (2012). Research on case teaching of software testing course with open source software. *Advances in Information Sciences and Service Sciences*, *4*(13).

Fylan, F. (2005). Semi-structured interviewing. *A handbook of research methods for clinical and health psychology*, 65–78.

Garland, R. (1991). The mid-point on a rating scale: Is it desirable. *Marketing bulletin*, *2*(1), 66–70.

Garousi, V., Coşkunçay, A., Betin-Can, A. & Demirörs, O. (2015). A survey of software engineering practices in Turkey. *Journal of Systems and Software*, *108*, 148–177. doi: 10.1016/j.jss.2015.06.036

Garousi, V. & Mäntylä, M. V. (2016). A systematic literature review of literature reviews in software testing. *Information and Software Technology*, *80*(November), 1339–1351. doi: 10.1016/j.infsof.2016.09.002

Garousi, V. & Zhi, J. (2013). A survey of software testing practices in Canada. *Journal of Systems and Software*, *86*(5), 1354–1376. doi: 10.1016/j.jss.2012.12.051

Hammond, M. & Wellington, J. J. (2013). *Research methods : the key concepts.* London ; New York : Routledge, 2013.

Hase, S. & Davis, L. (1999). From competence to capability: the implications for human resource development and management. *Graduate College of Management Papers*, 163.

Holtkamp, P., Jokinen, J. P. P. & Pawlowski, J. M. (2015). Soft competency requirements in requirements engineering, software design, implementation, and testing. *Journal of Systems and Software*, *101*, 136–146. doi: 10.1016/j.jss.2014.12.010

Hsieh, H.-F. & Shannon, S. E. (2005). The use of flipped classrooms to stimulate students' participation in an academic course in initial teacher education. *Nordic Journal of Digital Literacy*, *15*, 1277–1288. doi: 10.1177/1049732305276687

Iivonen, J. (2009). Identifying and characterizing highly performing testers–a case study in three software product companies. *Journal of Systems Integration*, *1*(2).

Iivonen, J., Mäntylä, M. V. & Itkonen, J. (2010). Characteristics of high performing testers. *IEEE/ACM International Symposium on Empirical Software Engineering and Measurement - ESEM*(C), 1. doi: 10.1145/1852786.1852862

Ikonen, A., Piironen, A., Saurén, K. & Lankinen, P. (2009). Cdio concept in challenge based learning. In *Proceedings of the 2009 workshop on embedded systems education* (pp. 27–32). New York, NY, USA: ACM. doi: 10.1145/1719010.1719016

ISTQB. (2016). ISTQB ® Worldwide Software Testing Practices Report.

Itkonen, J., Mäntylä, M. V. & Lassenius, C. (2009). How do testers do it? An exploratory study on manual testing practices. *2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009*, 494–497. doi: 10.1109/ESEM.2009.5314240

Jia, S. & Yang, C. (2013). Teaching software testing based on cdio. *World Transactions on Engineering and Technology Education*, *11*(4), 476-479.

John, J. (2009). Study on the Nature of Impact of Soft Skills Training Programme on the Soft Skills Development of Management Students. *Pacific Business Review*, *10*(12), 19–27.

Jones, E. L. (2001). Integrating testing into the curriculum — arsenic in small doses. *ACM SIGCSE Bulletin*, *33*(1), 337–341. doi: 10.1145/366413.364617

Joshi, G. & Desai, P. (2016, Dec). Building software testing skills in undergraduate students using spiral model approach. In *2016 ieee eighth international conference on technology for education (t4e)* (p. 244-245). doi: 10.1109/T4E.2016.061

Kaner, C., Bach, J. & Pettichord, B. (2002). *Lessons learned in software testing: a context-driven approach.* New York : Wiley, [2002].

Kaner, C. & Padmanabhan, S. (2007, July). Practice and transfer of learning in the teaching of software testing. In *20th conference on software engineering education training (cseet'07)* (pp. 157–166). doi: 10.1109/CSEET.2007.38

Kanij, T., Merkel, R. & Grundy, J. (2011). A Preliminary Study on Factors Affecting Software Testing Team Performance. *2011 International Symposium on Empirical Software Engineering and Measurement*(c), 359–362. doi: 10.1109/ESEM.2011.48

Kanij, T., Merkel, R. & Grundy, J. (2013). An empirical study of the effects of personality on software testing. *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*, 239–248. doi: 10.1109/CSEET.2013.6595255

Kanij, T., Merkel, R. & Grundy, J. (2014). A preliminary survey of factors affecting software testers. *Proceedings of the Australian Software Engineering Conference, ASWEC*, 180–189. doi: 10.1109/ASWEC.2014.32

Kanij, T., Merkel, R. & Grundy, J. (2015). An empirical investigation of personality traits of software testers. *Proceedings - 8th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2015*, 1–7. doi: 10.1109/CHASE.2015.7

Kannan, M. & Vivekanandan, K. (2012). A study on attrition among new entrants in software testing professionals. *International Journal of Computer Applications*, *53*(7), 23–29.

Kettunen, V., Kasurinen, J., Taipale, O. & Smolander, K. (2010). A study on agility and testing processes in software organizations. *Human Factors*, 231–240. doi: 10.1145/1831708.1831737

Kondracki, N. L., Wellman, N. S. & Amundson, D. R. (2002). Content analysis: review of methods and their applications in nutrition education. *Journal of nutrition education and behavior*, *34*(4), 224–230.

Krathwohl, D. (2016). Anderson and Krathwohl - Understanding the New Version of Bloom ' s Taxonomy The Cognitive Domain : Anderson and Krathwohl - Bloom ' s Taxonomy Revised. (1972).

Krutz, D. E., Malachowsky, S. A. & Reichlmayr, T. (2014). Using a real world project in a software testing course. In *Proceedings of the 45th acm technical symposium on computer science education* (pp. 49–54). doi: 10.1145/2538862.2538955

Lacher, L., Walia, G., Nygard, K., Fagerholm, F., Pagels, M. & Münch, J. (2015). A behavior marker for measuring non-technical skills of software professionals: An empirical study. *International Journal of Software Engineering and Knowledge Engineering*, *25*(9-10), 1733–1738. doi: 10.1142/S0218194015710084

Le Deist, F. D. & Winterton, J. (2005). What Is Competence? *Human Resource Development International*, *8*(1), 27–46. doi: 10.1080/1367886042000338227

Lee, J., Kang, S. & Lee, D. (2012). Survey on software testing practices. *IET Software*, *6*(3), 275. doi: 10.1049/iet-sen.2011.0066

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Martin, D., Rooksby, J., Rouncefield, M. & Sommerville, I. (2007). 'Good' organisational reasons for 'bad' software testing: An ethnographic study of testing in a small software company. *Proceedings - International Conference on Software Engineering*, 602–611. doi: 10.1109/ICSE.2007.1

Matturro, G. (2013). Soft skills in software engineering: A study of its demand by software companies in Uruguay. *2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2013 - Proceedings*, 133–136. doi: 10.1109/CHASE.2013.6614749

Matturro, G., Raschetti, F. & Fontán, C. (2015). Soft skills in software development teams: A survey of the points of view of team leaders and team members. *Proceedings - 8th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2015*, 101–104. doi: 10.1109/CHASE.2015.30

Meneely, A. & Lucidi, S. (2013). Vulnerability of the day: concrete demonstrations for software engineering undergraduates. In *Proceedings of the 2013 international conference on software engineering* (pp. 1154–1157).

Mtsweni, E. S., Hö, T. & Poll, J. A. V. D. (2016). Soft Skills for Software Project Team Members. *International Journalof Computer Theory and Engineering*, *8*(2), 1–6. doi: 10.7763/IJCTE.2016.V8.1035

Mulcahy, D. (2000). Turning the contradictions of competence: competency-based training and beyond. *Journal of vocational education and training*, *52*(2), 259–280.

Muller, T. & Friedenberg, D. (2011). Certified Tester Foundation Level Syllabus. *International Software Testing Qualifications Board*.

Nouman, M., Pervez, U., Hasan, O. & Saghar, K. (2016). Software testing: A survey and tutorial on white and black-box testing of C/C++ programs. *Proceedings - 2016 IEEE Region 10 Symposium, TENSYMP 2016*, 225–230. doi: 10.1109/TENCONSpring.2016.7519409

Papadopoulos, G. (2015). Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects. *Procedia - Social and Behavioral Sciences*, *175*, 455–463. doi: http://dx.doi.org/10.1016/j.sbspro.2015.01.1223

Pawlowski, J. M. & Holtkamp, P. (2012). Towards an internationalization of the information systems curriculum. *Multikonferenz Wirtschaftsinformatik 2012 - Tagungsband der MKWI 2012*, *2011*.

Petrova, K. & Medlin, D. (2008). Informing industry via academic research in ICT skill and capability development. *Business Web Strategy: Design, Alignment, and Application: Design, Alignment, and Application*.

Phelps, R., Hase, S. & Ellis, A. (2005). Competency, capability, complexity and computers: exploring a new model for conceptualising end-user computer education. *British Journal of Educational Technology*, *36*(1), 67–84.

Pradarelli, B., Nouet, P. & Latorre, L. (2016). Industrial test project oriented education. In *2016 ieee global engineering education conference (educon)* (p. 119-124).

Puleio, M. (2006). How not to do agile testing. In *Agile conference, 2006* (pp. 7–pp).

Rivera-Ibarra, J. G., Rodríguez-Jacobo, J. & Serrano-Vargas, M. A. (2010). Competency Framework for Software Engineers. *2010 23rd IEEE Conference on Software Engineering Education and Training*, 33–40. doi: 10.1109/CSEET.2010.21

Robles, M. M. (2012). Executive Perceptions of the Top 10 Soft Skills Needed in Today's Workplace. *Business Communication Quarterly*, *75*(4), 453–465. doi: 10.1177/1080569912460400

Rooksby, J., Rouncefield, M. & Sommerville, I. (2009). Testing in the wild: The Social and organisational dimensions of real world practice. *Computer Supported Cooperative Work*, *18*(5-6), 559–580. doi: 10.1007/s10606-009-9098-7

Runeson, P., Andersson, C. & Höst, M. (2003). Test processes in software product evolution - A qualitative survey on the state of practice. *Journal of Software Maintenance and Evolution*, *15*(1), 41–59. doi: 10.1002/smr.265

Runeson, P. & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, *14*(2), 131–164. doi: 10.1007/s10664-008-9102-8

Ryan, S. & O'connor, R. V. (2009). Development of a team measure for tacit knowledge in software development teams. *Journal of Systems and Software*, *82*(2), 229–240.

Sadraei, E., Aurum, A., Beydoun, G. & Paech, B. (2007). A field study of the requirements engineering practice in australian software industry. *Requirements Engineering*, *12*(3), 145–162.

Sedelmaier, Y. & Landes, D. (2014). Software engineering body of skills (SWEBOS). *IEEE Global Engineering Education Conference, EDUCON*(April), 395–401. doi: 10.1109/EDUCON.2014.6826125

Sengupta, B., Chandra, S. & Sinha, V. (2006). A Research Agenda for Distributed Software Development. *Icse*, 731–740. doi: 10.1145/1134285.1134402

Shah, H. & Harrold, M. J. (2010). Studying human and social aspects of testing in a service-based software company: Case study. *Proceedings - International Conference on Software Engineering*, 102–108. doi: 10.1145/1833310.1833327

Shah, H., Harrold, M. J. & Sinha, S. (2014). Global software testing under deadline pressure: Vendor-side experiences. *Information and Software Technology*, *56*, 6–19. doi: 10.1016/j.infsof.2013.04.005

Shanwen, Z. & Zhengguang, Z. (2012). Talent Training of Software Testing Based on University-Enterprise Cooperation. , *2*, 315–320.

Shaye, S. D. (2008). Transitioning a team to agile test methods. In *Agile, 2008. agile'08. conference* (pp. 470–477).

Shoaib, L., Nadeem, A. & Akbar, A. (2009). An empirical evaluation of the influence of human personality on exploratory software testing. *International Multitopic Conference*, 1–6. doi: 10.1109/INMIC.2009.5383088

Shuman, L. J., Besterfield-Sacre, M. & McGourty, J. (2005). The abet "professional skills"—can they be taught? can they be assessed? *Journal of engineering education*, *94*(1), 41–55.

Stolberg, S. (2009). Enabling agile testing through continuous integration. In *Agile conference, 2009. agile'09.* (pp. 369–374).

Tockey, S. (2015). Insanity, hiring, and the software industry. *Computer*, *48*(11), 96–101.

Vaismoradi, M., Turunen, H. & Bondas, T. (2013). Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. *Nursing & Health Sciences*, *15*(3), 398–405. doi: 10.1111/nhs.12048

Valentin, E., Carvalho, J. R. H. & Barreto, R. (2015). Rapid improvement of students' soft-skills based on an agile-process approach. *Proceedings - Frontiers in Education Conference, FIE*, *2014*. doi: 10.1109/FIE.2015.7344408

Valle, P. H. D., Barbosa, E. F. & Maldonado, J. C. (2015). CS curricula of the most relevant universities in Brazil and abroad: Perspective of software testing education. *2015 International Symposium on Computers in Education, SIIE 2015*, 62–68. doi: 10.1109/SIIE.2015.7451649

Vegas, S., Juristo, N. & Basili, V. (2006). Packaging experiences for improving testing technique selection. *Journal of Systems and Software*, *79*(11), 1606–1618. doi: 10.1016/j.jss.2006.02.049

Velde, C. (1999). An alternative conception of competence: implications for vocational education. *Journal of Vocational Education and Training*, *51*(3), 437–447.

Weyuker, E. J., Ostrand, T. J., Brophy, J. A. & Prasad, R. (2000). Clearing a career path for software testers. *IEEE Software*, *17*(2), 76–81. doi: 10.1109/52.841696

Whittaker, J. A. (2000). What is software testing? and why is it so hard? *IEEE software*, *17*(1), 70–79.

Whyte, G. & Mulder, D. L. (2011). Mitigating the Impact of Software Test Constraints on Software Testing Effectiveness. *Ejisecom*, *14*(2), 254–270.

Wikle, T. A. & Fagin, T. D. (2015). Hard and Soft Skills in Preparing GIS Professionals: Comparing Perceptions of Employers and Educators. *Transactions in GIS*, *19*(5), 641–652. doi: 10.1111/tgis.12126

Wilhelm, W. J. (2004). Determinants of moral reasoning: Academic factors, gender, richness-of-life experiences, and religious preferences. *Delta Pi Epsilon Journal*, *46*(2).

Winterton, J. (2009). Competence across Europe: highest common factor or lowest common denominator? *Journal of European Industrial Training*, *33*, 681–700. doi: 10.1108/03090590910993571

Wong, W. E., Bertolino, A., Debroy, V., Mathur, A., Offutt, J. & Vouk, M. (2011). Teaching Software Testing : Experiences , Lessons Learned and the Path Forward. *Software Engineering Education and Training (CSEE&T)*, 530–534.

Yilmaz, M., O'Connor, R. V. & Clarke, P. (2012). A Systematic Approach to the Comparison of Roles in the Software Development Processes. *Software Process Improvement and Capability Determination - Proceedings 12th International Conference on Process Improvement and Capability dEtermination in Software, Systems Engineering and Service Management, SPICE 2012*, *290*, 198–209. doi: 10.1007/978-3-642-30439-2

Yinnan, Z. & Xiaochi, W. (2011). *Implementation of Software Testing Course based on CDIO.* doi: 10.1109/ICCSE.2011.6028595

Yu, L., Xu, X., Liu, C. & Sheng, B. (2012). Using grounded theory to understand testing engineers' soft skills of third-party software testing centers. *ICSESS 2012 - Proceedings of 2012 IEEE 3rd International Conference on Software Engineering and Service Science*, 403–406. doi: 10.1109/ICSESS.2012.6269490

Appendix A

List of Abbreviations

AI Artificial Intelligence

AT Automation Tester

AUTEC Auckland University of Technology Ethics Committee

BA Business Analyst

BDD Behaviour-driven Development

CA Content Analysis

CDIO Conceive, Develop, Implement and Operate

CI/CD Continuous Integration and Continuous Delivery

DCA Directed Content Analysis

HR Human Resource

InT Integration Tester

ISTQB International Software Testing Qualifications Board

LR Literature Review

NZ New Zealand

PO Product Owner

PT Performance Tester

PTA Product Test Analyst

QA Quality Assurance

QDTM Question-Driven Teaching Method

RQs Research Questions

S&Cs Skills and Capabilities

SCA Summative Content Analysis

SD Software Development

SDP Software Development Process

ST Software Testing

SWEBOS Software Engineering Body of Skills

TA Test Analyst

TDD Test-driven Development

TE Test Engineer

TL Test Lead

TM Test Manager

TTA Technical Test Analyst

VBA Visual Basic for Applications

Appendix B

Data for Literature Review

B.1 Highly Ranked Skills and Capabilities of Software Testers Found in Literature Review

B.2 Other Important Skills and Capabilities of Software Testers Found in Literature Review

Interpersonal Skills: the ability to interact with people in comfortable way (Ahmed, Capretz & Campbell, 2012), or build personal relationships with others (Rivera-Ibarra et al., 2010). It can help to create an easy working environment (Rivera-Ibarra et al., 2010).

Programming knowledge: the knowledge can obtained from education and industry experience (Deak, 2014b). It helps testers communicating better with developers because they know the language of developers (Deak, 2014b). Also, some testing roles, such as automation tester, require intensive programming (Jones, 2001).

Attention to details: or detailed oriented, the ability to make even little things done correctly (John, 2009). Armour (2005) stated that "testing, more than other activities

Table B.1: Existing Studies on Skills and Capabilities for Software Testers

Literature	Area	Number of Respondents	Main Research Approaches
Martin et al.,(2007)	STs	–	Experiments
Beer & Ramler, (2008)	SDTs	3 companies	Multiple case study
Itkonen, Mäntylä & Lassenius, (2009)	STs	four software development companies	Observations Interviews
Iivonen et al., (2010)	STs	12 participants (3 companies)	Interviews
Rivera-Ibarra et al., (2010)	SEs	9 groups of master's students	observations interviews
Kettunen et al., (2010)	STs	12 orgnisations	Observations
Kanij, Merkel & Grundy, (2011)	STs	100 groups	Questionnaire
Lee, Kang & Lee, (2012)	STs	22 companies 20 experts	Questionnaire
Ahmed et al., (2013)	SDTs	650 job adverts 165 in testing	Surveys
Matturro, (2013)	SEs	60 testing adverts 43 with soft skills	Surveys
Kanij, (2014)	STs	24 groups	Online Surveys
Deak, (2014b)	STs	11 testers in 4 companies	Interviews
Holtkamp et al., (2015)	SEs	32 participants with different roles	Questionnaire
Matturro et al., (2015)	SDTs	35 software engineers	Interviews
Deak, Stålhane, & Sindre (2016)	STs	12 companies with 36 practitioners	Interviews
Mtsweni, Hörne & Poll (2016)	SDTs	6 QAs	Questionnaire

Table B.2: Frequencies of Top Skills and Capabilities in Literature

Skill Name	Frequency	Skill Name	Frequency
Domain Knowledge	7	Test Automation	2
Communication Skills	6	IT Background	2
Teamwork	5	Analytical, Problem-solving	2
Experience in Testing Techniques	3	Good language /English skill	2
Experience in Testing Tools	2	Patience	2

in software development, is about discovery". Detail-oriented tester can review the artefacts (specifications, codes, etc.) concisely and thoroughly, not missing tiny relevant information. It is an advantage to check unhappy paths and discover more defects (Deak, 2014b).

Curiosity: "the desire to learn or know about anything"[1]. It motivates testers to "improve understanding of the product" and "come up with some unusual testing scenarios"(Deak, 2014b).

Autonomous: the ability to work independently. It means one can carry out tasks with minimal supervision (Ahmed, Capretz & Campbell, 2012), or do things automatically without the reminder of other people (John, 2009). People with this capability know what to do.

Adaptability: the ability to accept or adapt to changes and new technologies (Ahmed, Capretz & Campbell, 2012). It can help testers doing well with unforeseeable situations (John, 2009). Things are changing all the time, so we have to get used to new things.

B.3 Anderson and Krathwohl's Taxonomy on Cognitive Skills

[1] http://www.dictionary.com/browse/curiosity

Table B.3: Anderson and Krathwohl's Taxonomy on Cognitive Skills

Skill Level	Key Words
Remembering	Decite, reproduce, recall, definitions, facts, list
Understanding	Comprehend, understand, explain, interpret, rewrite, compare, summarise, exemplify
Applying	Implement, apply, execute, carry out, construct, use, develop, programme, demonstrate, translate
Analysing	Analyse, break down, compare, contrast, deconstruct, differentiate, organise, attribute, distinguish, identify, outline
Evaluating	Evaluate, assess, judge, estimate, validate, conclude, make decision, choose, judge
Creating	Create, synthesis, reorganise, new, creative, resolution, design, problem solving

Appendix C

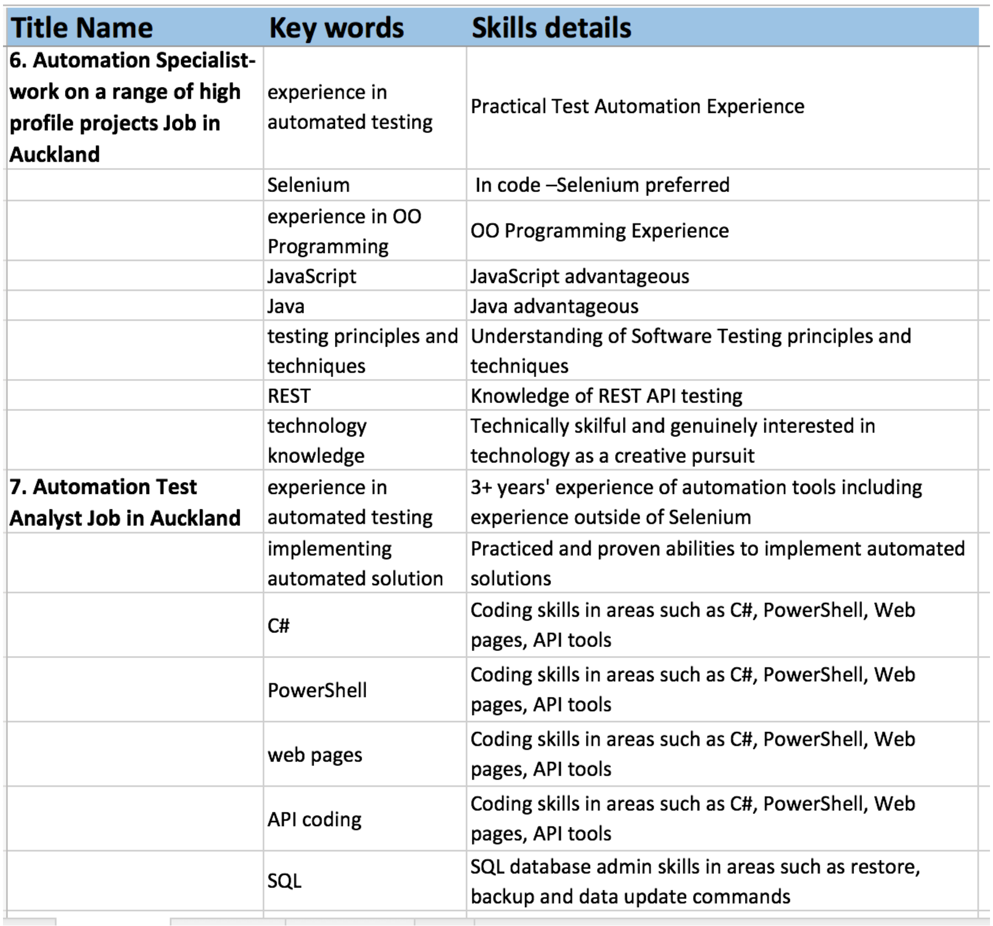
Data for Job Adverts

C.1 Table Generated in Job Adverts Analysing

C.2 Synonyms and Frequencies of the Skills and Capabilities in Job Adverts

C.3 Synonyms and Frequencies of the Responsibilities in Job Adverts

Table C.1: Manually Extracted Skills and Capabilities from Job Samples

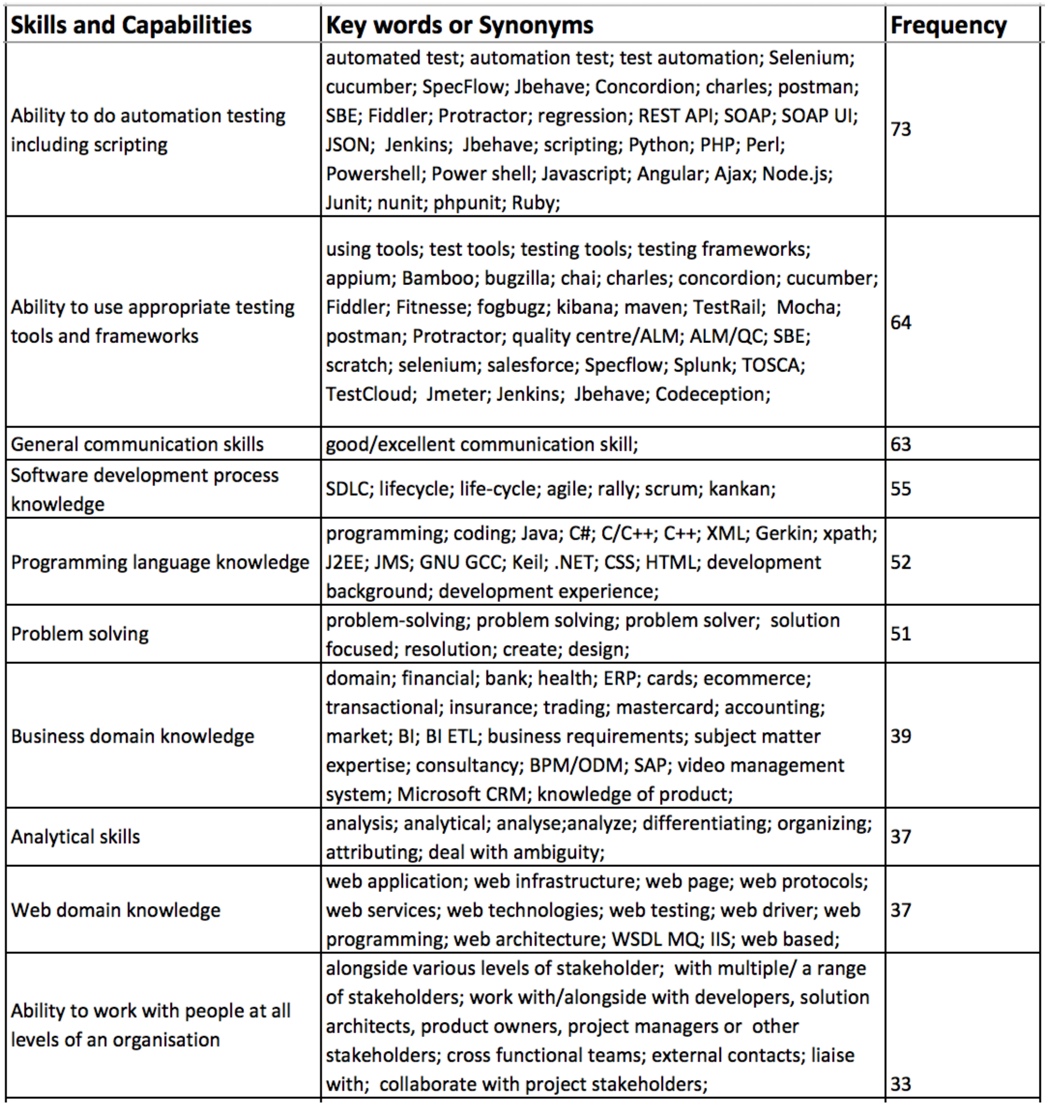


Title Name	Key words	Skills details
6. Automation Specialist-work on a range of high profile projects Job in Auckland	experience in automated testing	Practical Test Automation Experience
	Selenium	In code –Selenium preferred
	experience in OO Programming	OO Programming Experience
	JavaScript	JavaScript advantageous
	Java	Java advantageous
	testing principles and techniques	Understanding of Software Testing principles and techniques
	REST	Knowledge of REST API testing
	technology knowledge	Technically skilful and genuinely interested in technology as a creative pursuit
7. Automation Test Analyst Job in Auckland	experience in automated testing	3+ years' experience of automation tools including experience outside of Selenium
	implementing automated solution	Practiced and proven abilities to implement automated solutions
	C#	Coding skills in areas such as C#, PowerShell, Web pages, API tools
	PowerShell	Coding skills in areas such as C#, PowerShell, Web pages, API tools
	web pages	Coding skills in areas such as C#, PowerShell, Web pages, API tools
	API coding	Coding skills in areas such as C#, PowerShell, Web pages, API tools
	SQL	SQL database admin skills in areas such as restore, backup and data update commands

Table C.2: Expressions of Different Roles in Job Adverts

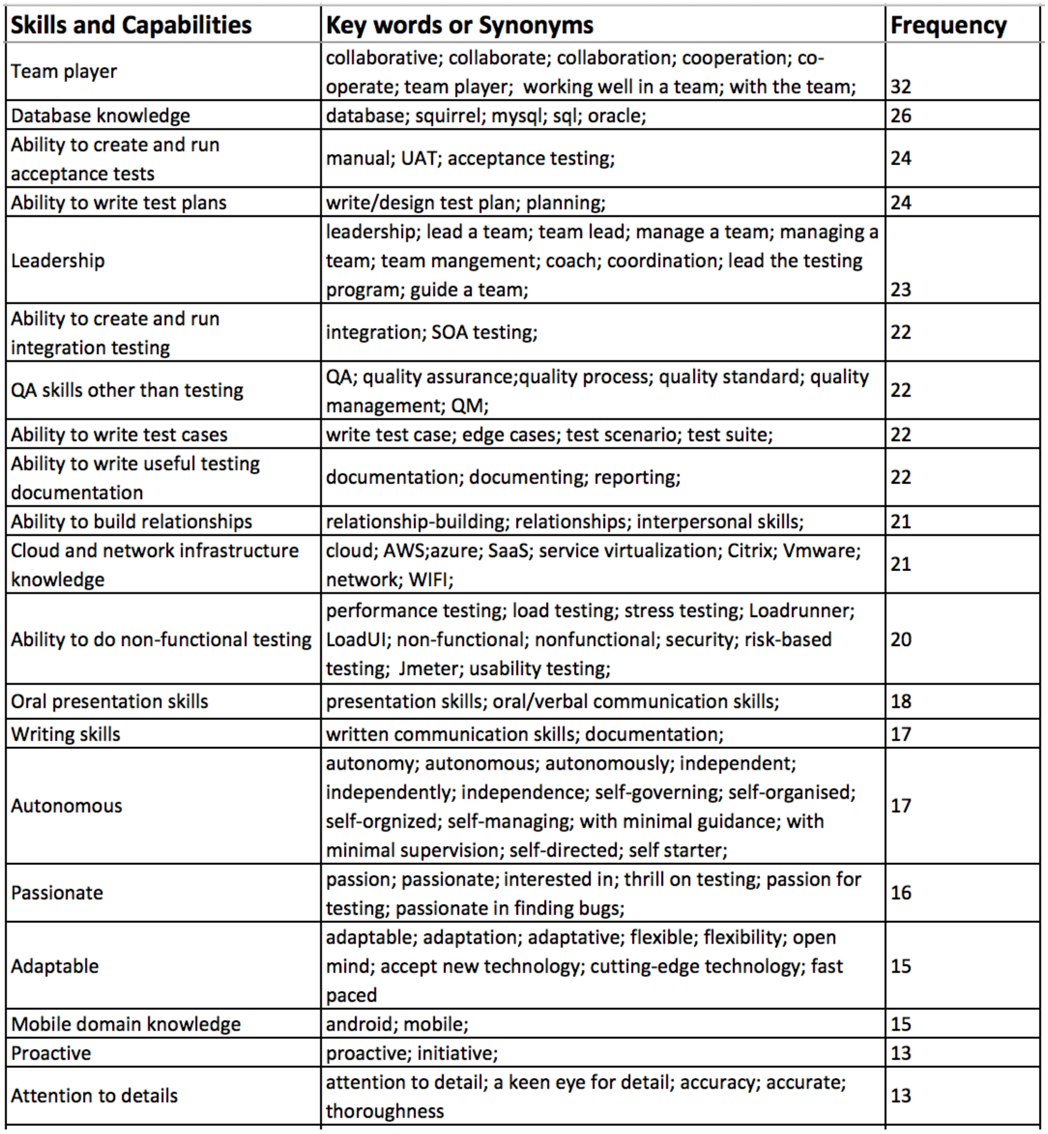
Role Name	Similar Expressions
Test Analyst (TA)	(Software) Test Analyst (A08, A14); Quality Assurance Analyst (A09); Test Manager; (Software) Test Engineer (A15); Software Tester (A16); Automation & Functional Tester (A23); (Senior) Certification and Test Engineer (A29); (Senior) QA Test Analyst (A30); Test and Release Lead (A36); Test Lead (A37); Test Manager (A38); Lead QA Engineer (B18); Lead Tester (B19); Quality Engineer (Software Tester) (B28); (Senior) Embedded Test Engineer (B37); (Senior) Software Test Analyst (Manual & Functional) (B43); (Senior) Test Engineer (B47); Software / Devices Tester (B51); Test Analyst - Automation, Wealth Management, Investment (B56); Test Analyst –intermediate to senior (B65); Test and Release Lead (B66); Test Director (B69);
Quality Assurance (QA)	National Lead for Quality (B22); (Senior) Quality Analyst (B41); (Senior) Quality Business Analyst(B42); Software QA Engineer (A13);
Automation Tester (AT)	Automation Tester (A04); Automation Test Analyst (A01); Software Engineer - C#/Automation(A12); Automation Test Lead (A24,A25); (Senior) Automation Test Engineer (A28); Technical Test Analyst (A32); (Senior) Test Automation Engineer (A35); (Senior) Automation Engineer (B4,B35); (Senior) Automation Mobile Test Analyst (B05); Automation Specialist (B06); Mobile Automation Tester (B20); Selenium Test Automation Expert (B30); Senior Automated Test Manager (B33); (Senior) Development Engineer in Test (B36); Test Automation Analyst / Senior QA Engineer/ Web C# / Automation (B67);
Product Test Analyst (PTA)	Dynamics CRM Test Analyst (A06, A07); SAP Test Analyst (A10); Lead Product Tester -Exploratory and Automation Job (A27); Test Analyst - Cards Switching; Web Applications Tester (B74);
Integration Tester (IT)	Integration Test Lead (A26); (Senior) Systems Integration Tester (A31); Systems Integration Test Lead (B52);
Performance Tester (PA)	Performance and Volume Test Lead (P&V Test Lead) (B22); Performance Senior Technical Test Analyst (B23); Senior Performance Engineer (B38);Senior Performance Test Engineer (B39); Technical Tester expertise in load and performance testing (B54);
Test Engineer (TE)	Firmware Test Engineer (B12, B13);

Table C.3: Synonyms and Frequencies of Skills and Capabilities in Job Adverts (1)



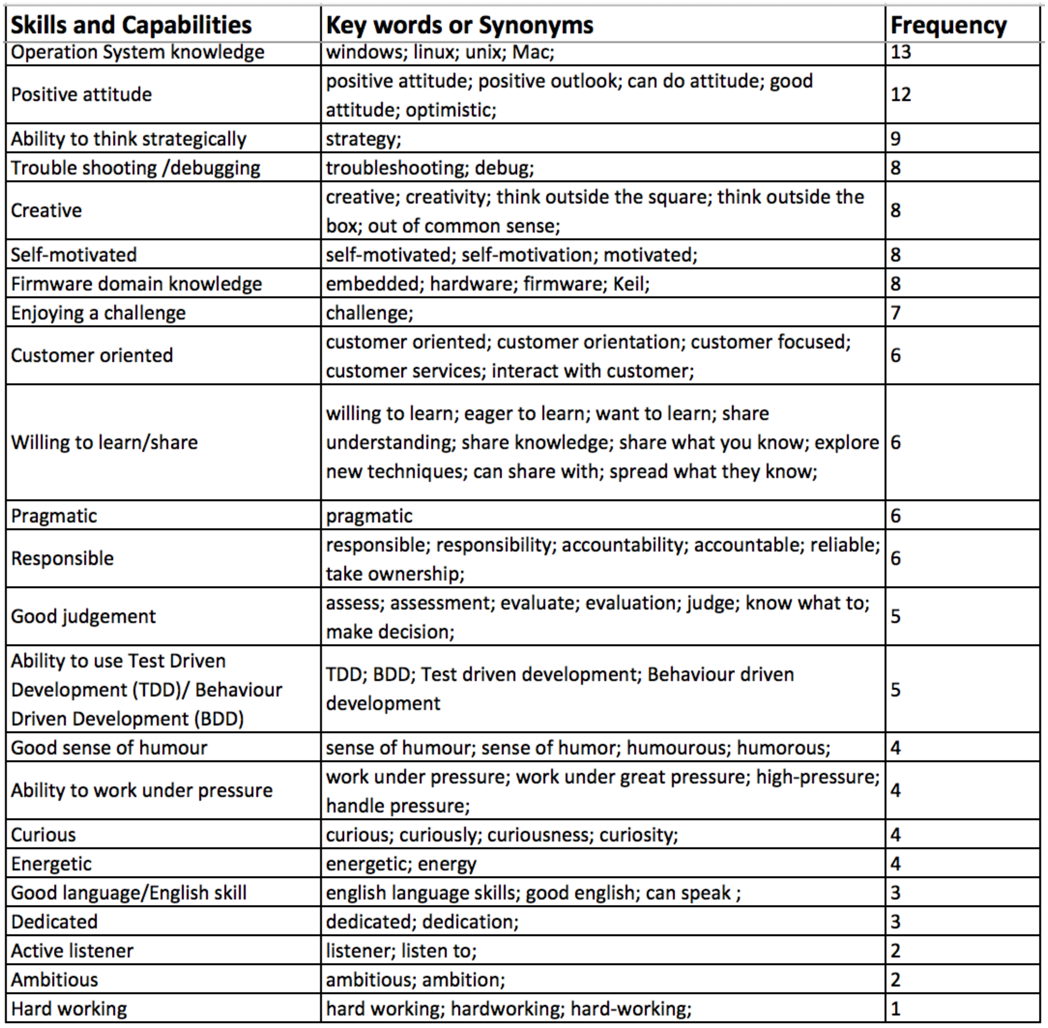
Skills and Capabilities	Key words or Synonyms	Frequency
Ability to do automation testing including scripting	automated test; automation test; test automation; Selenium; cucumber; SpecFlow; Jbehave; Concordion; charles; postman; SBE; Fiddler; Protractor; regression; REST API; SOAP; SOAP UI; JSON; Jenkins; Jbehave; scripting; Python; PHP; Perl; Powershell; Power shell; Javascript; Angular; Ajax; Node.js; Junit; nunit; phpunit; Ruby;	73
Ability to use appropriate testing tools and frameworks	using tools; test tools; testing tools; testing frameworks; appium; Bamboo; bugzilla; chai; charles; concordion; cucumber; Fiddler; Fitnesse; fogbugz; kibana; maven; TestRail; Mocha; postman; Protractor; quality centre/ALM; ALM/QC; SBE; scratch; selenium; salesforce; Specflow; Splunk; TOSCA; TestCloud; Jmeter; Jenkins; Jbehave; Codeception;	64
General communication skills	good/excellent communication skill;	63
Software development process knowledge	SDLC; lifecycle; life-cycle; agile; rally; scrum; kankan;	55
Programming language knowledge	programming; coding; Java; C#; C/C++; C++; XML; Gerkin; xpath; J2EE; JMS; GNU GCC; Keil; .NET; CSS; HTML; development background; development experience;	52
Problem solving	problem-solving; problem solving; problem solver; solution focused; resolution; create; design;	51
Business domain knowledge	domain; financial; bank; health; ERP; cards; ecommerce; transactional; insurance; trading; mastercard; accounting; market; BI; BI ETL; business requirements; subject matter expertise; consultancy; BPM/ODM; SAP; video management system; Microsoft CRM; knowledge of product;	39
Analytical skills	analysis; analytical; analyse;analyze; differentiating; organizing; attributing; deal with ambiguity;	37
Web domain knowledge	web application; web infrastructure; web page; web protocols; web services; web technologies; web testing; web driver; web programming; web architecture; WSDL MQ; IIS; web based;	37
Ability to work with people at all levels of an organisation	alongside various levels of stakeholder; with multiple/ a range of stakeholders; work with/alongside with developers, solution architects, product owners, project managers or other stakeholders; cross functional teams; external contacts; liaise with; collaborate with project stakeholders;	33

Table C.4: Synonyms and Frequencies of Skills and Capabilities in Job Adverts (2)



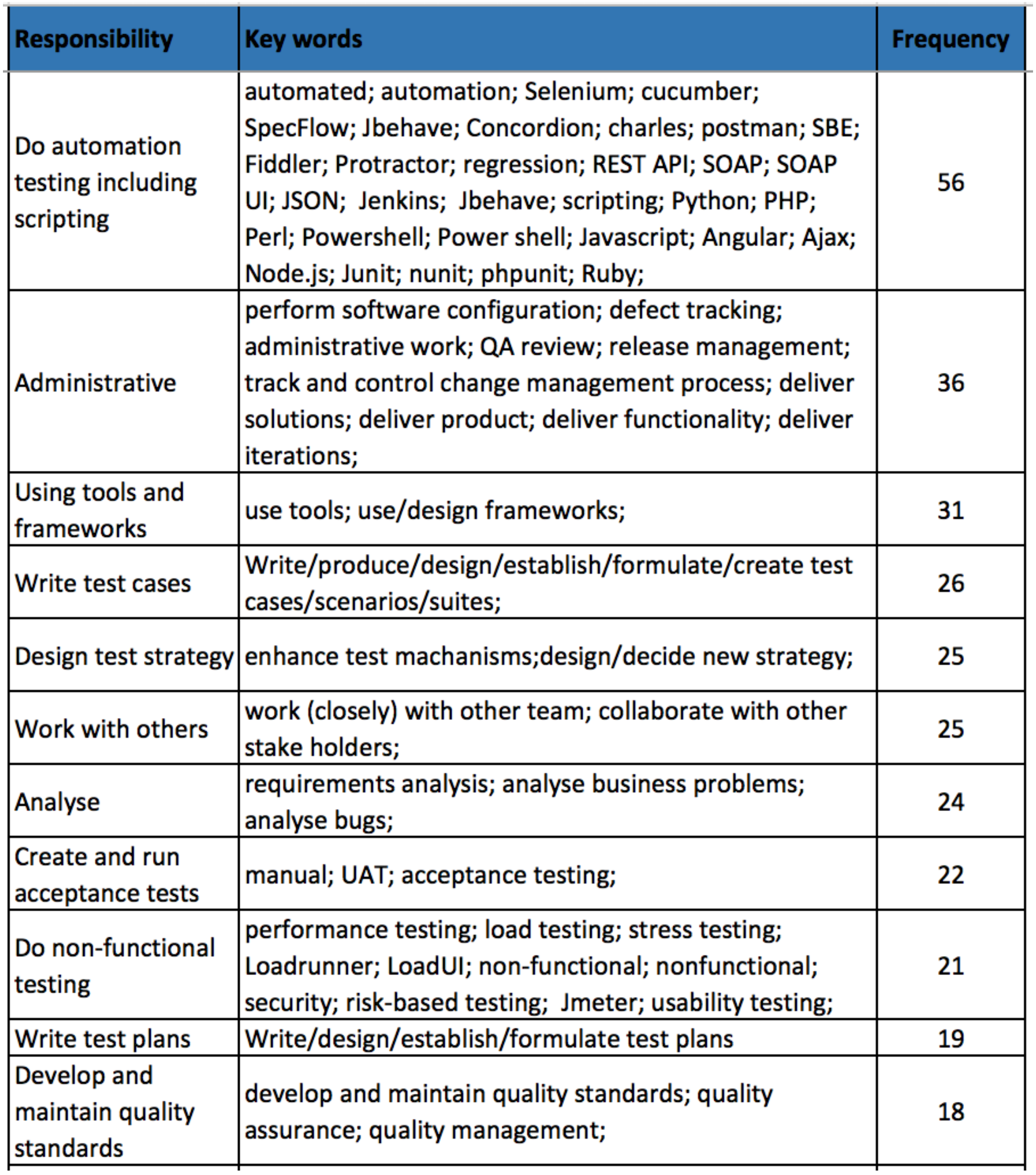
Skills and Capabilities	Key words or Synonyms	Frequency
Team player	collaborative; collaborate; collaboration; cooperation; co-operate; team player; working well in a team; with the team;	32
Database knowledge	database; squirrel; mysql; sql; oracle;	26
Ability to create and run acceptance tests	manual; UAT; acceptance testing;	24
Ability to write test plans	write/design test plan; planning;	24
Leadership	leadership; lead a team; team lead; manage a team; managing a team; team mangement; coach; coordination; lead the testing program; guide a team;	23
Ability to create and run integration testing	integration; SOA testing;	22
QA skills other than testing	QA; quality assurance;quality process; quality standard; quality management; QM;	22
Ability to write test cases	write test case; edge cases; test scenario; test suite;	22
Ability to write useful testing documentation	documentation; documenting; reporting;	22
Ability to build relationships	relationship-building; relationships; interpersonal skills;	21
Cloud and network infrastructure knowledge	cloud; AWS;azure; SaaS; service virtualization; Citrix; Vmware; network; WIFI;	21
Ability to do non-functional testing	performance testing; load testing; stress testing; Loadrunner; LoadUI; non-functional; nonfunctional; security; risk-based testing; Jmeter; usability testing;	20
Oral presentation skills	presentation skills; oral/verbal communication skills;	18
Writing skills	written communication skills; documentation;	17
Autonomous	autonomy; autonomous; autonomously; independent; independently; independence; self-governing; self-organised; self-orgnized; self-managing; with minimal guidance; with minimal supervision; self-directed; self starter;	17
Passionate	passion; passionate; interested in; thrill on testing; passion for testing; passionate in finding bugs;	16
Adaptable	adaptable; adaptation; adaptative; flexible; flexibility; open mind; accept new technology; cutting-edge technology; fast paced	15
Mobile domain knowledge	android; mobile;	15
Proactive	proactive; initiative;	13
Attention to details	attention to detail; a keen eye for detail; accuracy; accurate; thoroughness	13

Table C.5: Synonyms and Frequencies of Skills and Capabilities in Job Adverts (3)



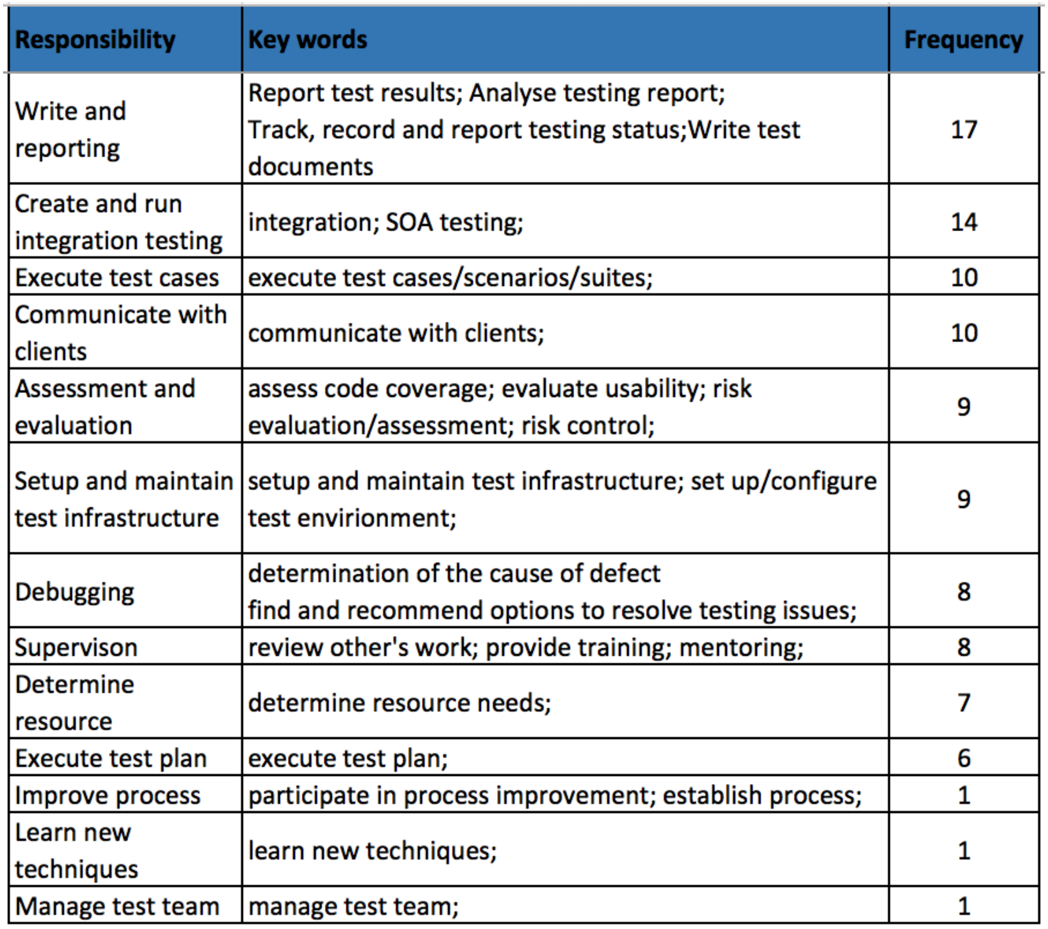
Skills and Capabilities	Key words or Synonyms	Frequency
Operation System knowledge	windows; linux; unix; Mac;	13
Positive attitude	positive attitude; positive outlook; can do attitude; good attitude; optimistic;	12
Ability to think strategically	strategy;	9
Trouble shooting /debugging	troubleshooting; debug;	8
Creative	creative; creativity; think outside the square; think outside the box; out of common sense;	8
Self-motivated	self-motivated; self-motivation; motivated;	8
Firmware domain knowledge	embedded; hardware; firmware; Keil;	8
Enjoying a challenge	challenge;	7
Customer oriented	customer oriented; customer orientation; customer focused; customer services; interact with customer;	6
Willing to learn/share	willing to learn; eager to learn; want to learn; share understanding; share knowledge; share what you know; explore new techniques; can share with; spread what they know;	6
Pragmatic	pragmatic	6
Responsible	responsible; responsibility; accountability; accountable; reliable; take ownership;	6
Good judgement	assess; assessment; evaluate; evaluation; judge; know what to; make decision;	5
Ability to use Test Driven Development (TDD)/ Behaviour Driven Development (BDD)	TDD; BDD; Test driven development; Behaviour driven development	5
Good sense of humour	sense of humour; sense of humor; humourous; humorous;	4
Ability to work under pressure	work under pressure; work under great pressure; high-pressure; handle pressure;	4
Curious	curious; curiously; curiousness; curiosity;	4
Energetic	energetic; energy	4
Good language/English skill	english language skills; good english; can speak ;	3
Dedicated	dedicated; dedication;	3
Active listener	listener; listen to;	2
Ambitious	ambitious; ambition;	2
Hard working	hard working; hardworking; hard-working;	1

Table C.6: Synonyms and Frequencies of Responsibilities in Job Adverts (1)



Responsibility	Key words	Frequency
Do automation testing including scripting	automated; automation; Selenium; cucumber; SpecFlow; Jbehave; Concordion; charles; postman; SBE; Fiddler; Protractor; regression; REST API; SOAP; SOAP UI; JSON; Jenkins; Jbehave; scripting; Python; PHP; Perl; Powershell; Power shell; Javascript; Angular; Ajax; Node.js; Junit; nunit; phpunit; Ruby;	56
Administrative	perform software configuration; defect tracking; administrative work; QA review; release management; track and control change management process; deliver solutions; deliver product; deliver functionality; deliver iterations;	36
Using tools and frameworks	use tools; use/design frameworks;	31
Write test cases	Write/produce/design/establish/formulate/create test cases/scenarios/suites;	26
Design test strategy	enhance test machanisms;design/decide new strategy;	25
Work with others	work (closely) with other team; collaborate with other stake holders;	25
Analyse	requirements analysis; analyse business problems; analyse bugs;	24
Create and run acceptance tests	manual; UAT; acceptance testing;	22
Do non-functional testing	performance testing; load testing; stress testing; Loadrunner; LoadUI; non-functional; nonfunctional; security; risk-based testing; Jmeter; usability testing;	21
Write test plans	Write/design/establish/formulate test plans	19
Develop and maintain quality standards	develop and maintain quality standards; quality assurance; quality management;	18

Table C.7: Synonyms and Frequencies of Responsibilities in Job Adverts (2)



Responsibility	Key words	Frequency
Write and reporting	Report test results; Analyse testing report; Track, record and report testing status;Write test documents	17
Create and run integration testing	integration; SOA testing;	14
Execute test cases	execute test cases/scenarios/suites;	10
Communicate with clients	communicate with clients;	10
Assessment and evaluation	assess code coverage; evaluate usability; risk evaluation/assessment; risk control;	9
Setup and maintain test infrastructure	setup and maintain test infrastructure; set up/configure test envirionment;	9
Debugging	determination of the cause of defect find and recommend options to resolve testing issues;	8
Supervison	review other's work; provide training; mentoring;	8
Determine resource	determine resource needs;	7
Execute test plan	execute test plan;	6
Improve process	participate in process improvement; establish process;	1
Learn new techniques	learn new techniques;	1
Manage test team	manage test team;	1

Appendix D

Design of Interview

D.1 Interview Questions

Q1. What is the main industry sector your organisation is in (e.g. banking, health, insurance, bespoke software)?

Q2. Please describe the software development process that the testers work in.

Q3. Would you describe the process as Agile or not?

Q4. How many testers does your company/organisation have? (How many in a development team?)

Q5. What is a typical ratio of developers to testers in your organisation?

Q6. What specialist software testing roles are there in your organisation?

Q7. What is your official job title?

Q8. Do you think of yourself as a junior, intermediate or senior tester?

Q9. How many years have you been a tester?

Table D.1: Grid of Communication Skills

Interviewee No: __________ Role Name: ____________

Communication Skills	**Importance (1-5)**	**Comments**
Oral presentation skills		
Writing skills		
Persuasive		
Active listener		
Empathetic to others		
Good language/English skill		

Q10. What role did you have before you were a tester?

Q11. What formal training in testing did you get before your current testing job?

Q12. What skills or capabilities that you have developed do you think have helped YOU to be a good tester?

Q13. What general skills and capabilities do you think are the most important ones for any testing role?

Q14. What are the most important differences in skills and capabilities between junior and intermediate testers?

Q15. What are the most important differences in skills and capabilities between intermediate and senior or lead testers?

Q16. What skills do you think you would need to develop to become a (a) test lead (b) test manager?

Q17. Grid Forms

Table D.2: Grid of Social Skills

Interviewee No: ____________ Role Name: _______________

Social Skills	**Importance (1-5)**	**Comments**
Team player		
Leadership		
Ability to build relationships		
Ability to work with people at all levels of an organisation		

Table D.3: Grid of Cognitive Skills

Interviewee No: _________ Role Name: __________

Cognitive Skills	**Importance (1-5)**	**Comments**
Critical thinking		
Good judgement		
Problem Solving		
Analytical skills		
Trouble shooting /debugging		
Creative		
Ability to think strategically		
Fast Learner		

Table D.4: Grid of Testing Skills

Interviewee No:______________ Role Name:_______________

Testing Skills	**Importance (1-5)**	**Comments**
Ability to create and run acceptance tests		
Ability to create and run integration testing		
Ability to do automation test including scripting		
Ability to do non-functional testing		
Ability to use Test Driven Development (TDD)/ Behaviour Driven Development (BDD)		
QA skills other than testing		
Ability to use appropriate testing tools and frameworks		
Ability to write test plans		
Ability to write test cases		
Ability to write useful testing documentation		

Table D.5: Grid of Other Knowledge

Interviewee No: _____________ Role Name: ________________

Other Knowledge	**Importance (1-5)**	**Comments**
Business domain knowledge		
Technical domain knowledge		
Software development process Knowledge		
Database knowledge		
Programming language Knowledge		

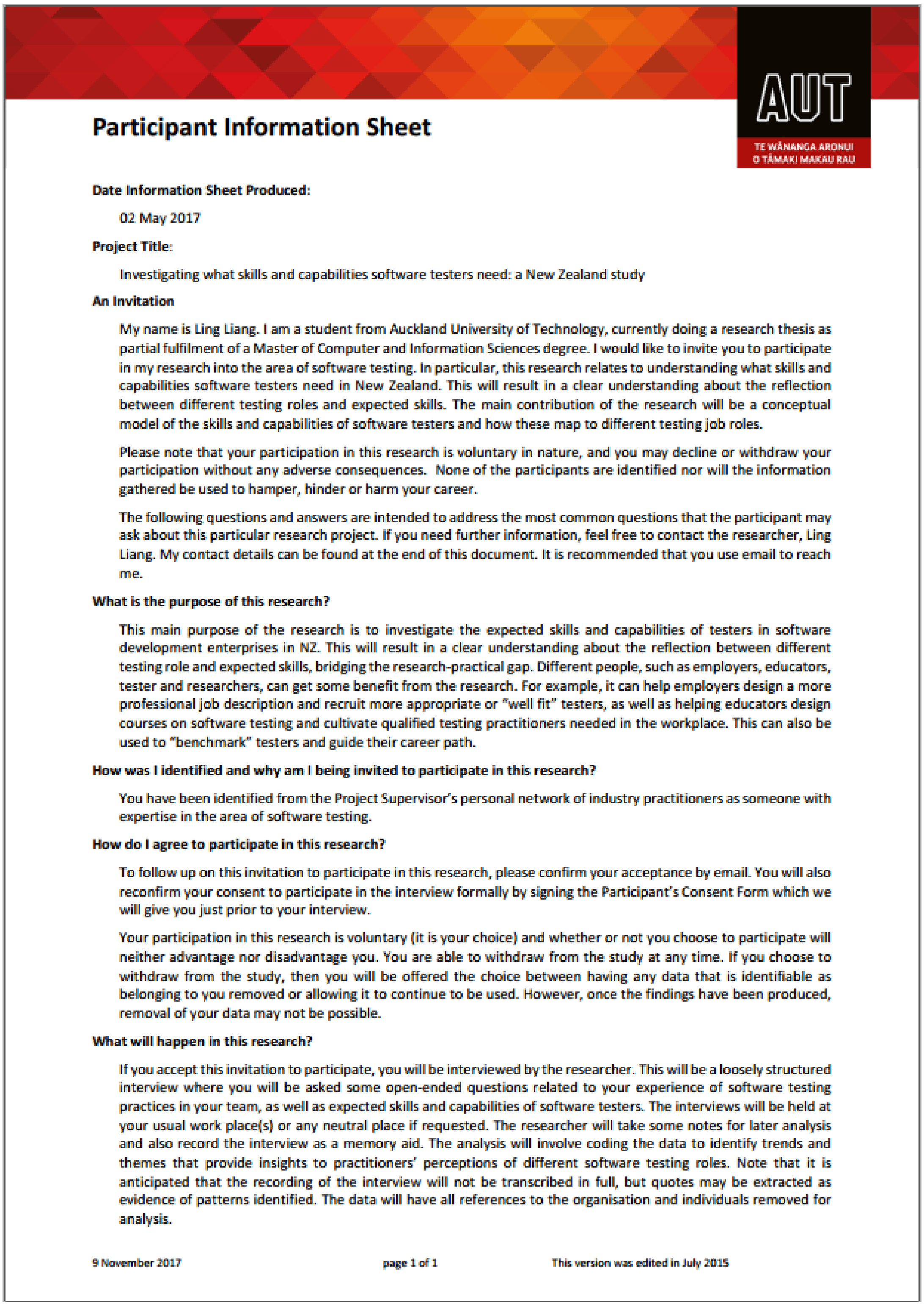
D.2 Participant Information Sheet

D.3 Content Form

Table D.6: Grid of Personal Qualities

Interviewee No: ______________ Role Name: _________________

Personal Qualities	**Importance (1-5)**	**Comments**
Passionate		
Adaptable		
Proactive		
Autonomous		
Attention to details		
Positive attitude		
Enjoying a challenge		
Customer oriented		
Hard working		
Willing to learn/share		
Good sense of humour		
Ability to work under pressure		
Curious		
Dedicated		
Pragmatic		
Responsible		
Self-motivated		
Energetic		
Ambitious		



AUT
TE WĀNANGA ARONUI
O TĀMAKI MAKAU RAU

Participant Information Sheet

Date Information Sheet Produced:

02 May 2017

Project Title:

Investigating what skills and capabilities software testers need: a New Zealand study

An Invitation

My name is Ling Liang. I am a student from Auckland University of Technology, currently doing a research thesis as partial fulfilment of a Master of Computer and Information Sciences degree. I would like to invite you to participate in my research into the area of software testing. In particular, this research relates to understanding what skills and capabilities software testers need in New Zealand. This will result in a clear understanding about the reflection between different testing roles and expected skills. The main contribution of the research will be a conceptual model of the skills and capabilities of software testers and how these map to different testing job roles.

Please note that your participation in this research is voluntary in nature, and you may decline or withdraw your participation without any adverse consequences. None of the participants are identified nor will the information gathered be used to hamper, hinder or harm your career.

The following questions and answers are intended to address the most common questions that the participant may ask about this particular research project. If you need further information, feel free to contact the researcher, Ling Liang. My contact details can be found at the end of this document. It is recommended that you use email to reach me.

What is the purpose of this research?

This main purpose of the research is to investigate the expected skills and capabilities of testers in software development enterprises in NZ. This will result in a clear understanding about the reflection between different testing role and expected skills, bridging the research-practical gap. Different people, such as employers, educators, tester and researchers, can get some benefit from the research. For example, it can help employers design a more professional job description and recruit more appropriate or "well fit" testers, as well as helping educators design courses on software testing and cultivate qualified testing practitioners needed in the workplace. This can also be used to "benchmark" testers and guide their career path.

How was I identified and why am I being invited to participate in this research?

You have been identified from the Project Supervisor's personal network of industry practitioners as someone with expertise in the area of software testing.

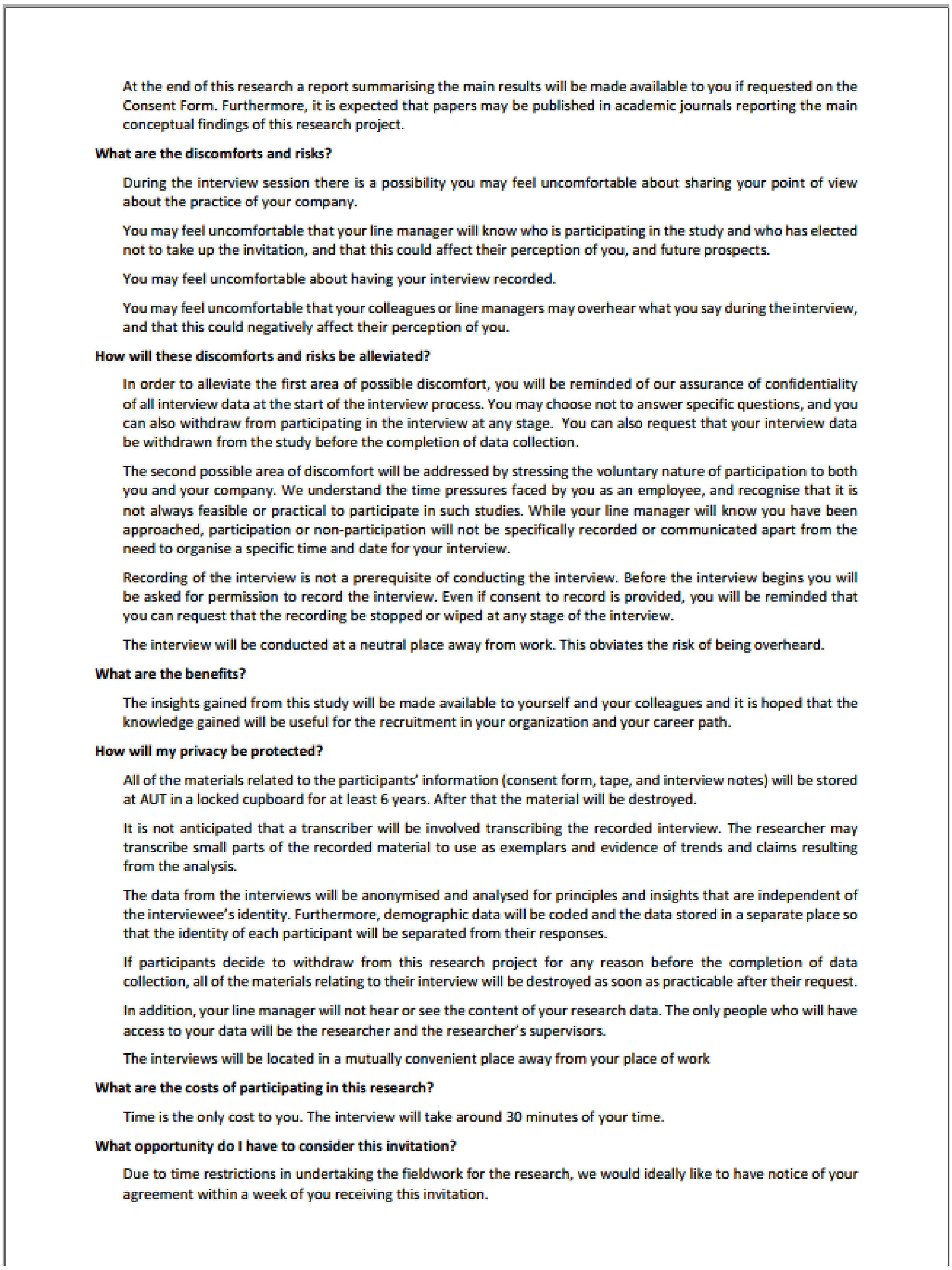
How do I agree to participate in this research?

To follow up on this invitation to participate in this research, please confirm your acceptance by email. You will also reconfirm your consent to participate in the interview formally by signing the Participant's Consent Form which we will give you just prior to your interview.

Your participation in this research is voluntary (it is your choice) and whether or not you choose to participate will neither advantage nor disadvantage you. You are able to withdraw from the study at any time. If you choose to withdraw from the study, then you will be offered the choice between having any data that is identifiable as belonging to you removed or allowing it to continue to be used. However, once the findings have been produced, removal of your data may not be possible.

What will happen in this research?

If you accept this invitation to participate, you will be interviewed by the researcher. This will be a loosely structured interview where you will be asked some open-ended questions related to your experience of software testing practices in your team, as well as expected skills and capabilities of software testers. The interviews will be held at your usual work place(s) or any neutral place if requested. The researcher will take some notes for later analysis and also record the interview as a memory aid. The analysis will involve coding the data to identify trends and themes that provide insights to practitioners' perceptions of different software testing roles. Note that it is anticipated that the recording of the interview will not be transcribed in full, but quotes may be extracted as evidence of patterns identified. The data will have all references to the organisation and individuals removed for analysis.



At the end of this research a report summarising the main results will be made available to you if requested on the Consent Form. Furthermore, it is expected that papers may be published in academic journals reporting the main conceptual findings of this research project.

What are the discomforts and risks?

During the interview session there is a possibility you may feel uncomfortable about sharing your point of view about the practice of your company.

You may feel uncomfortable that your line manager will know who is participating in the study and who has elected not to take up the invitation, and that this could affect their perception of you, and future prospects.

You may feel uncomfortable about having your interview recorded.

You may feel uncomfortable that your colleagues or line managers may overhear what you say during the interview, and that this could negatively affect their perception of you.

How will these discomforts and risks be alleviated?

In order to alleviate the first area of possible discomfort, you will be reminded of our assurance of confidentiality of all interview data at the start of the interview process. You may choose not to answer specific questions, and you can also withdraw from participating in the interview at any stage. You can also request that your interview data be withdrawn from the study before the completion of data collection.

The second possible area of discomfort will be addressed by stressing the voluntary nature of participation to both you and your company. We understand the time pressures faced by you as an employee, and recognise that it is not always feasible or practical to participate in such studies. While your line manager will know you have been approached, participation or non-participation will not be specifically recorded or communicated apart from the need to organise a specific time and date for your interview.

Recording of the interview is not a prerequisite of conducting the interview. Before the interview begins you will be asked for permission to record the interview. Even if consent to record is provided, you will be reminded that you can request that the recording be stopped or wiped at any stage of the interview.

The interview will be conducted at a neutral place away from work. This obviates the risk of being overheard.

What are the benefits?

The insights gained from this study will be made available to yourself and your colleagues and it is hoped that the knowledge gained will be useful for the recruitment in your organization and your career path.

How will my privacy be protected?

All of the materials related to the participants' information (consent form, tape, and interview notes) will be stored at AUT in a locked cupboard for at least 6 years. After that the material will be destroyed.

It is not anticipated that a transcriber will be involved transcribing the recorded interview. The researcher may transcribe small parts of the recorded material to use as exemplars and evidence of trends and claims resulting from the analysis.

The data from the interviews will be anonymised and analysed for principles and insights that are independent of the interviewee's identity. Furthermore, demographic data will be coded and the data stored in a separate place so that the identity of each participant will be separated from their responses.

If participants decide to withdraw from this research project for any reason before the completion of data collection, all of the materials relating to their interview will be destroyed as soon as practicable after their request.

In addition, your line manager will not hear or see the content of your research data. The only people who will have access to your data will be the researcher and the researcher's supervisors.

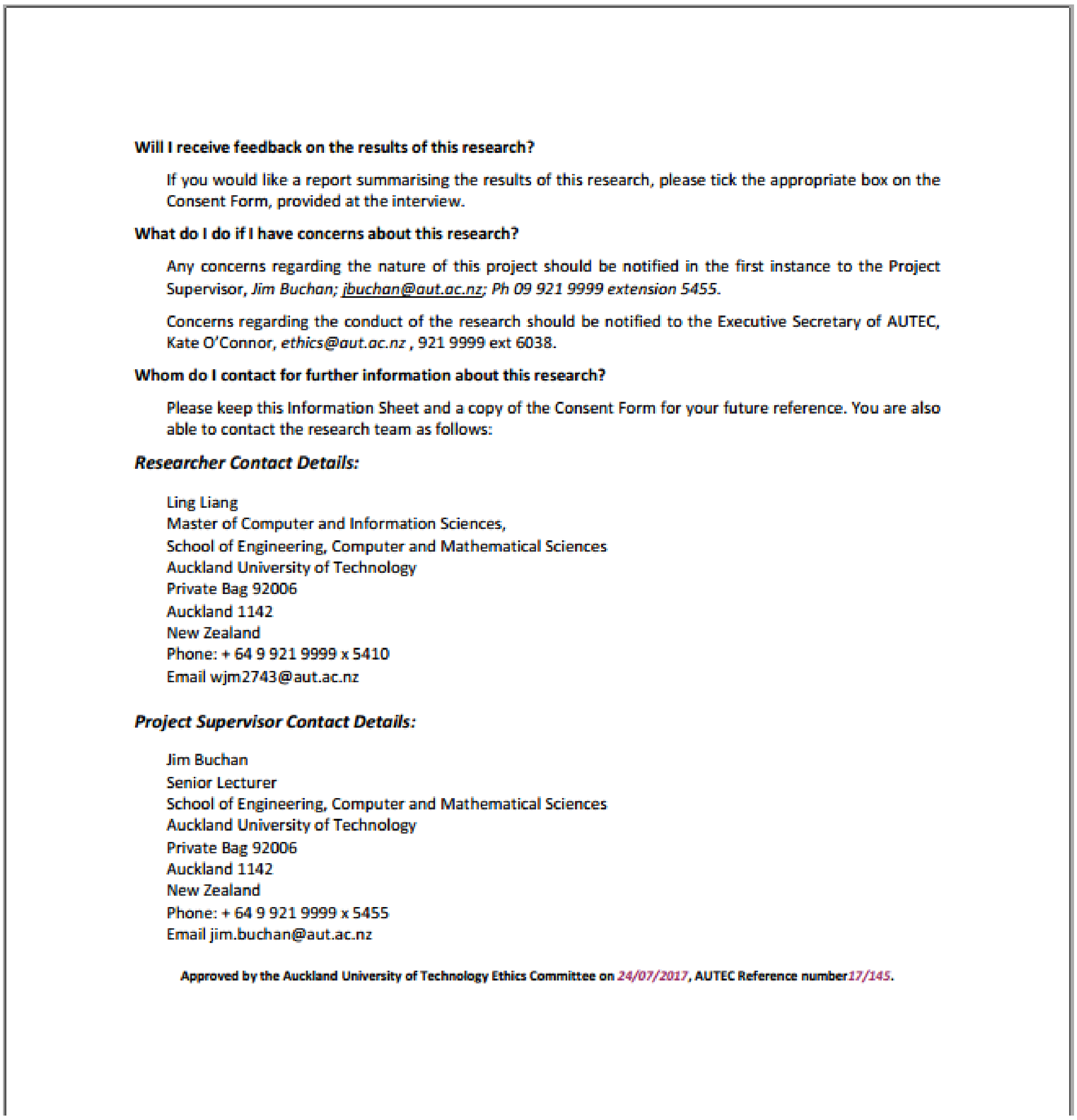
The interviews will be located in a mutually convenient place away from your place of work

What are the costs of participating in this research?

Time is the only cost to you. The interview will take around 30 minutes of your time.

What opportunity do I have to consider this invitation?

Due to time restrictions in undertaking the fieldwork for the research, we would ideally like to have notice of your agreement within a week of you receiving this invitation.



Will I receive feedback on the results of this research?

If you would like a report summarising the results of this research, please tick the appropriate box on the Consent Form, provided at the interview.

What do I do if I have concerns about this research?

Any concerns regarding the nature of this project should be notified in the first instance to the Project Supervisor, *Jim Buchan; jbuchan@aut.ac.nz; Ph 09 921 9999 extension 5455.*

Concerns regarding the conduct of the research should be notified to the Executive Secretary of AUTEC, Kate O'Connor, *ethics@aut.ac.nz* , 921 9999 ext 6038.

Whom do I contact for further information about this research?

Please keep this Information Sheet and a copy of the Consent Form for your future reference. You are also able to contact the research team as follows:

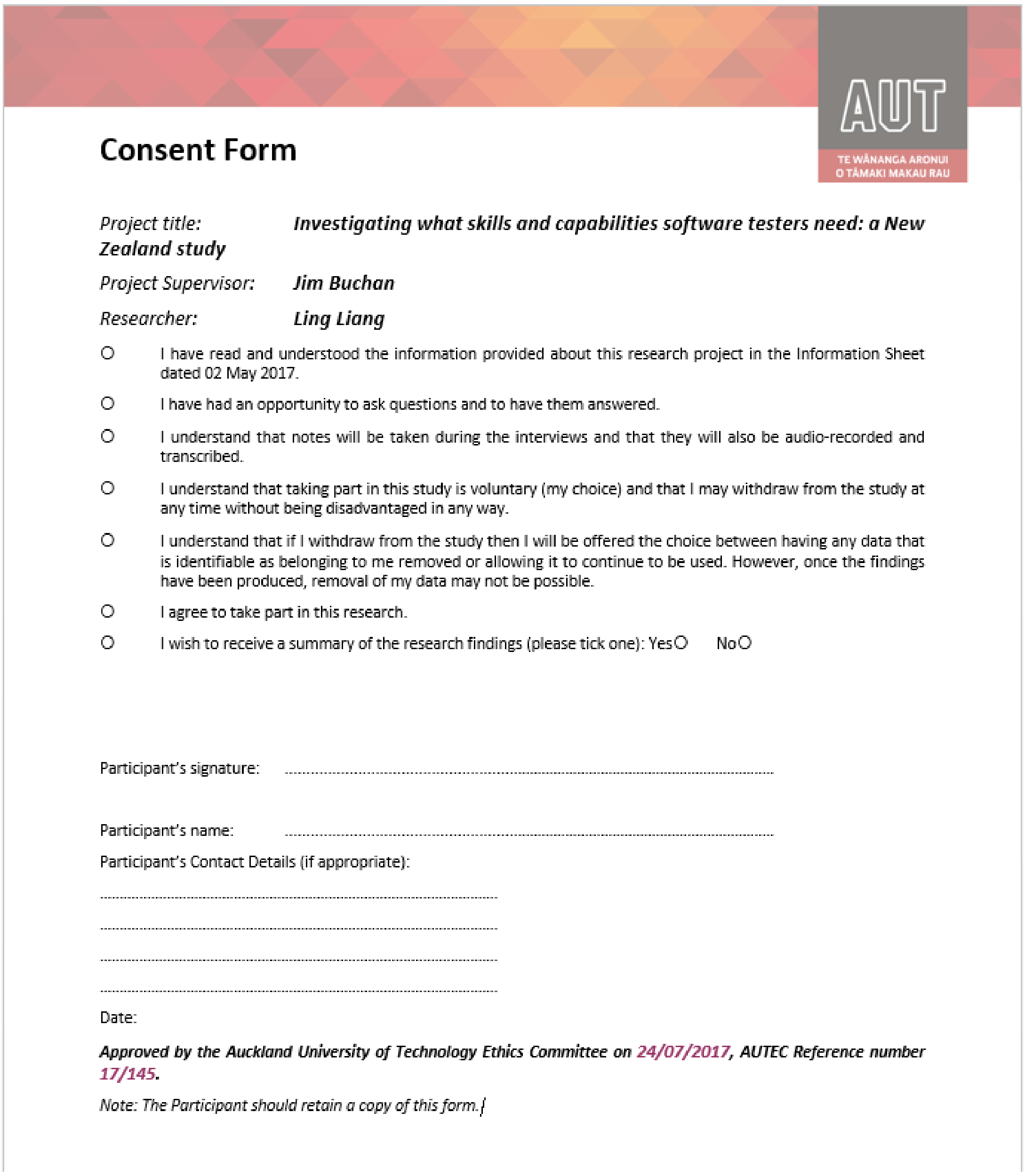
Researcher Contact Details:

Ling Liang
Master of Computer and Information Sciences,
School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology
Private Bag 92006
Auckland 1142
New Zealand
Phone: + 64 9 921 9999 x 5410
Email wjm2743@aut.ac.nz

Project Supervisor Contact Details:

Jim Buchan
Senior Lecturer
School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology
Private Bag 92006
Auckland 1142
New Zealand
Phone: + 64 9 921 9999 x 5455
Email jim.buchan@aut.ac.nz

Approved by the Auckland University of Technology Ethics Committee on *24/07/2017*, AUTEC Reference number*17/145*.



AUT
TE WĀNANGA ARONUI
O TĀMAKI MAKAU RAU

Consent Form

Project title: ***Investigating what skills and capabilities software testers need: a New Zealand study***

Project Supervisor: ***Jim Buchan***

Researcher: ***Ling Liang***

○ I have read and understood the information provided about this research project in the Information Sheet dated 02 May 2017.

○ I have had an opportunity to ask questions and to have them answered.

○ I understand that notes will be taken during the interviews and that they will also be audio-recorded and transcribed.

○ I understand that taking part in this study is voluntary (my choice) and that I may withdraw from the study at any time without being disadvantaged in any way.

○ I understand that if I withdraw from the study then I will be offered the choice between having any data that is identifiable as belonging to me removed or allowing it to continue to be used. However, once the findings have been produced, removal of my data may not be possible.

○ I agree to take part in this research.

○ I wish to receive a summary of the research findings (please tick one): Yes○ No○

Participant's signature: ....

Participant's name: ....

Participant's Contact Details (if appropriate):

....

....

....

....

Date:

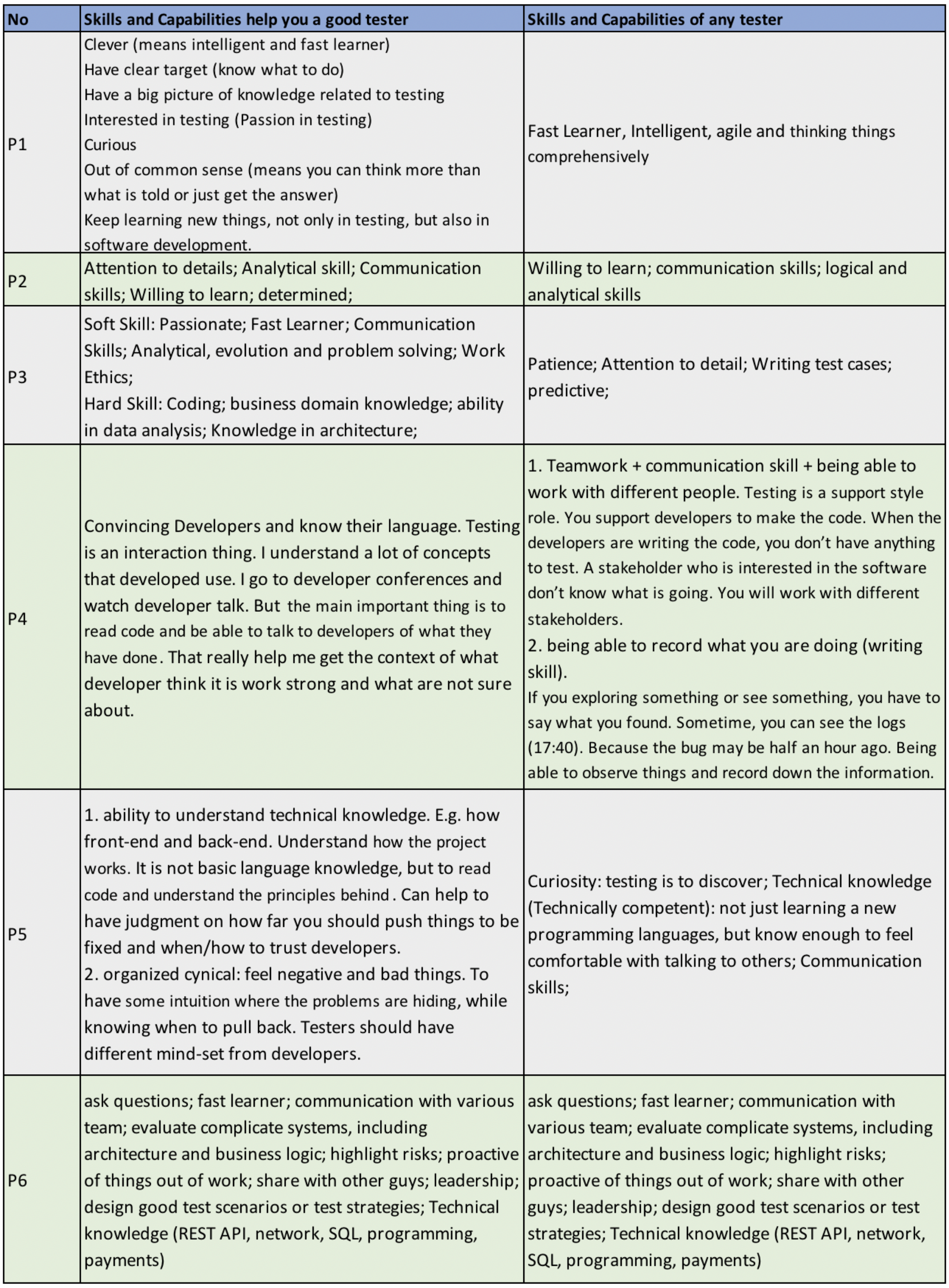
Approved by the Auckland University of Technology Ethics Committee on 24/07/2017, AUTEC Reference number 17/145.

Note: The Participant should retain a copy of this form.

Appendix E

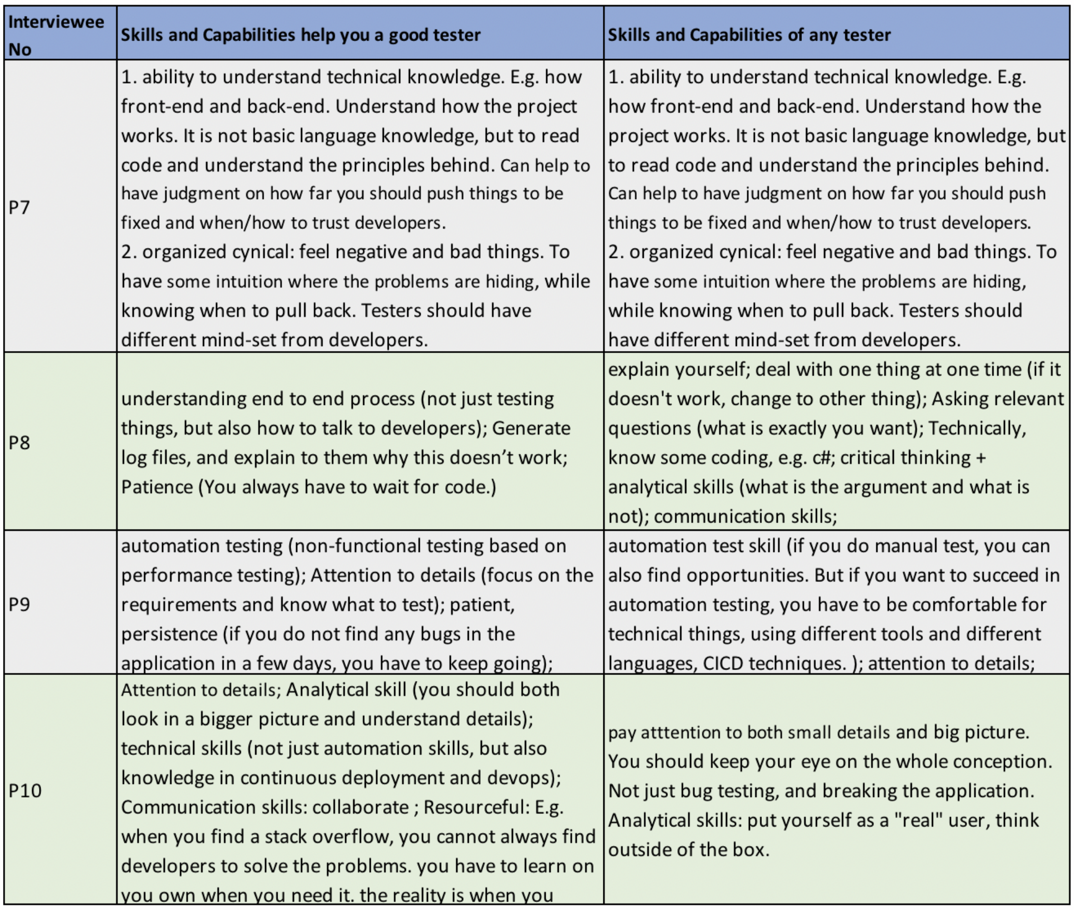
Data Obtained from Interview Questions

Table E.1: Highlighted Skills and Capabilities Obtained from Interview Questions Q12-13



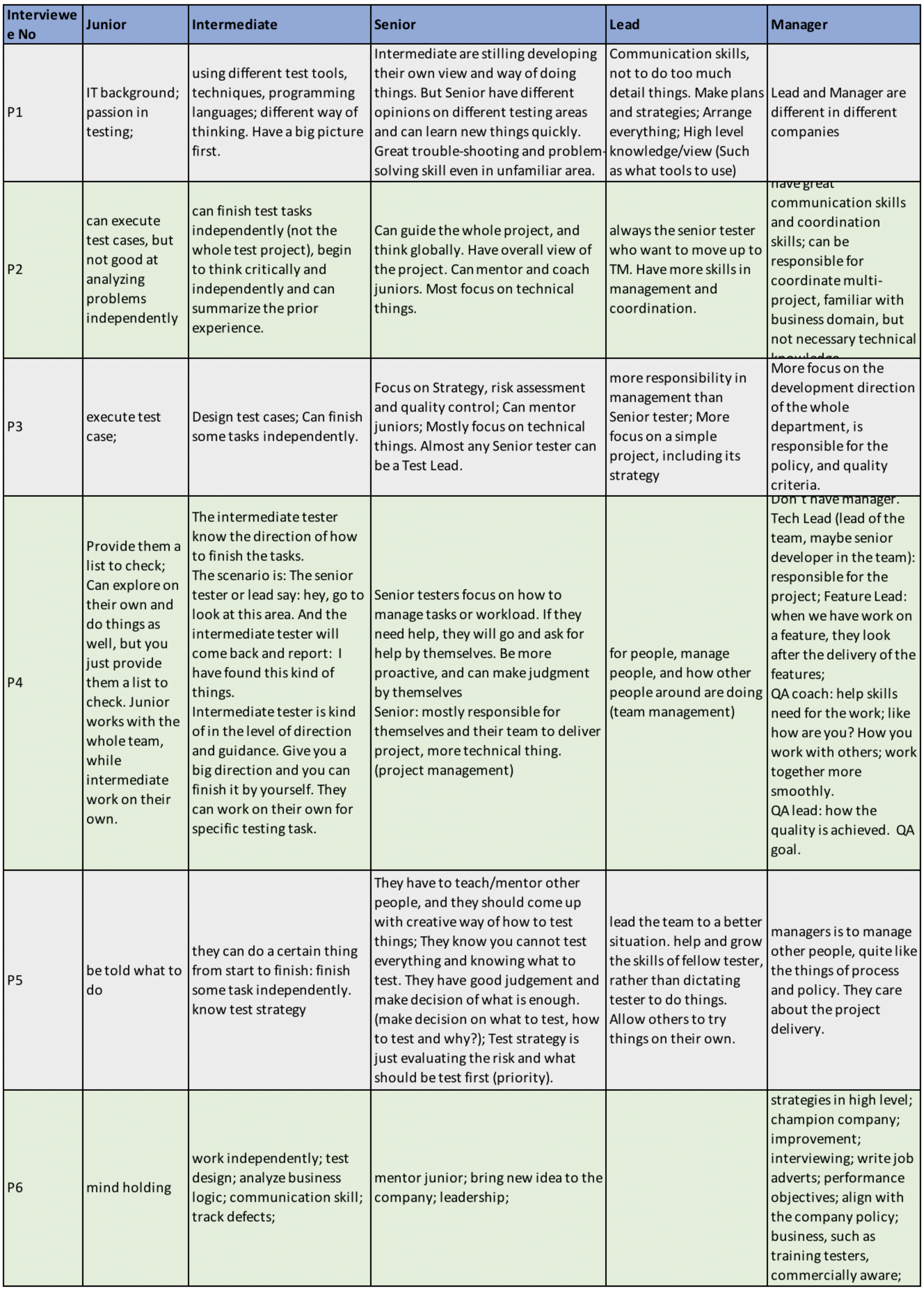
No	Skills and Capabilities help you a good tester	Skills and Capabilities of any tester
P1	Clever (means intelligent and fast learner) Have clear target (know what to do) Have a big picture of knowledge related to testing Interested in testing (Passion in testing) Curious Out of common sense (means you can think more than what is told or just get the answer) Keep learning new things, not only in testing, but also in software development.	Fast Learner, Intelligent, agile and thinking things comprehensively
P2	Attention to details; Analytical skill; Communication skills; Willing to learn; determined;	Willing to learn; communication skills; logical and analytical skills
P3	Soft Skill: Passionate; Fast Learner; Communication Skills; Analytical, evolution and problem solving; Work Ethics; Hard Skill: Coding; business domain knowledge; ability in data analysis; Knowledge in architecture;	Patience; Attention to detail; Writing test cases; predictive;
P4	Convincing Developers and know their language. Testing is an interaction thing. I understand a lot of concepts that developed use. I go to developer conferences and watch developer talk. But the main important thing is to read code and be able to talk to developers of what they have done. That really help me get the context of what developer think it is work strong and what are not sure about.	1. Teamwork + communication skill + being able to work with different people. Testing is a support style role. You support developers to make the code. When the developers are writing the code, you don't have anything to test. A stakeholder who is interested in the software don't know what is going. You will work with different stakeholders. 2. being able to record what you are doing (writing skill). If you exploring something or see something, you have to say what you found. Sometime, you can see the logs (17:40). Because the bug may be half an hour ago. Being able to observe things and record down the information.
P5	1. ability to understand technical knowledge. E.g. how front-end and back-end. Understand how the project works. It is not basic language knowledge, but to read code and understand the principles behind. Can help to have judgment on how far you should push things to be fixed and when/how to trust developers. 2. organized cynical: feel negative and bad things. To have some intuition where the problems are hiding, while knowing when to pull back. Testers should have different mind-set from developers.	Curiosity: testing is to discover; Technical knowledge (Technically competent): not just learning a new programming languages, but know enough to feel comfortable with talking to others; Communication skills;
P6	ask questions; fast learner; communication with various team; evaluate complicate systems, including architecture and business logic; highlight risks; proactive of things out of work; share with other guys; leadership; design good test scenarios or test strategies; Technical knowledge (REST API, network, SQL, programming, payments)	ask questions; fast learner; communication with various team; evaluate complicate systems, including architecture and business logic; highlight risks; proactive of things out of work; share with other guys; leadership; design good test scenarios or test strategies; Technical knowledge (REST API, network, SQL, programming, payments)

Table E.1: Highlighted Skills and Capabilities Obtained from Interview Questions Q12-13



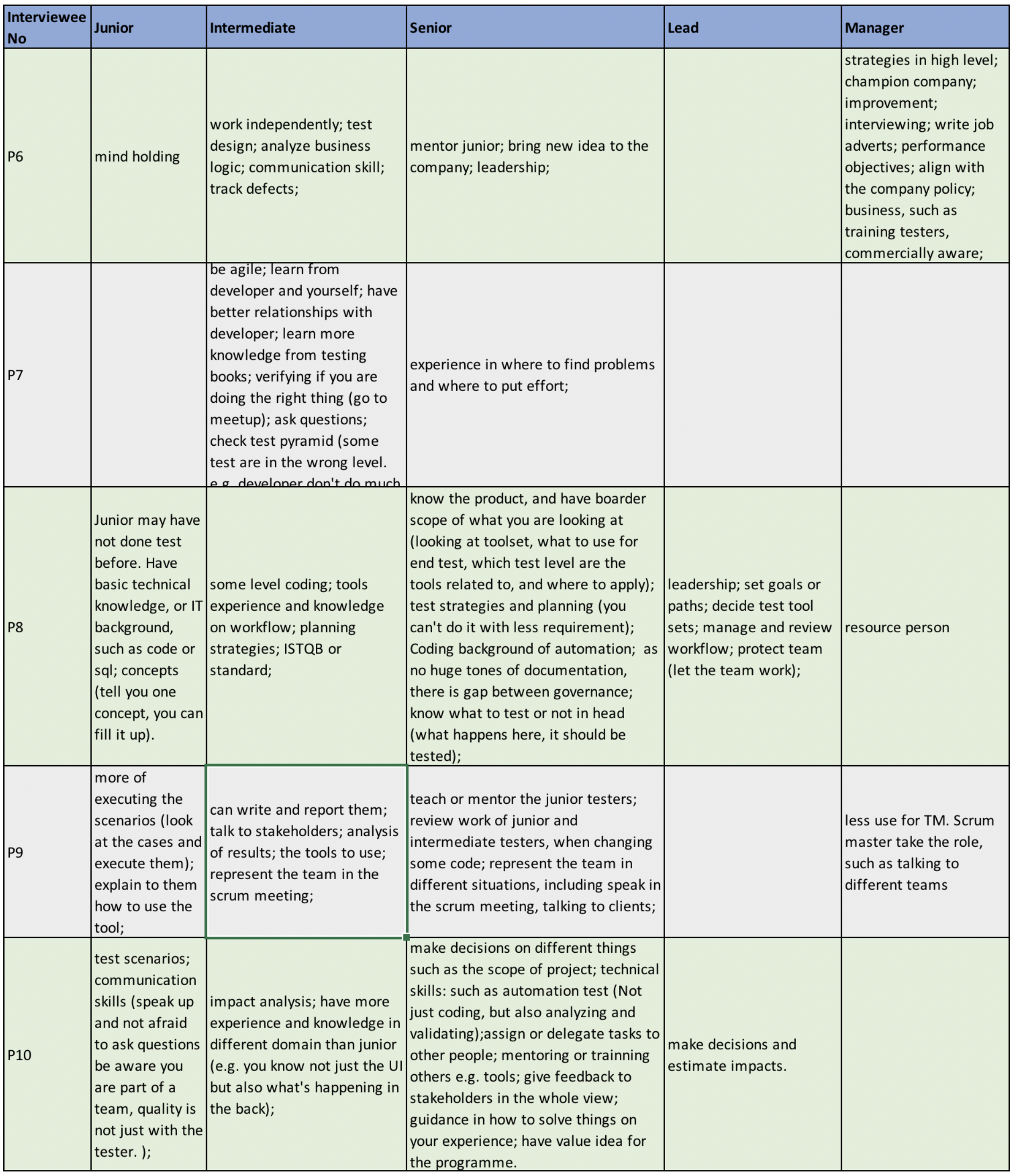
Interviewee No	Skills and Capabilities help you a good tester	Skills and Capabilities of any tester
P7	1. ability to understand technical knowledge. E.g. how front-end and back-end. Understand how the project works. It is not basic language knowledge, but to read code and understand the principles behind. Can help to have judgment on how far you should push things to be fixed and when/how to trust developers. 2. organized cynical: feel negative and bad things. To have some intuition where the problems are hiding, while knowing when to pull back. Testers should have different mind-set from developers.	1. ability to understand technical knowledge. E.g. how front-end and back-end. Understand how the project works. It is not basic language knowledge, but to read code and understand the principles behind. Can help to have judgment on how far you should push things to be fixed and when/how to trust developers. 2. organized cynical: feel negative and bad things. To have some intuition where the problems are hiding, while knowing when to pull back. Testers should have different mind-set from developers.
P8	understanding end to end process (not just testing things, but also how to talk to developers); Generate log files, and explain to them why this doesn't work; Patience (You always have to wait for code.)	explain yourself; deal with one thing at one time (if it doesn't work, change to other thing); Asking relevant questions (what is exactly you want); Technically, know some coding, e.g. c#; critical thinking + analytical skills (what is the argument and what is not); communication skills;
P9	automation testing (non-functional testing based on performance testing); Attention to details (focus on the requirements and know what to test); patient, persistence (if you do not find any bugs in the application in a few days, you have to keep going);	automation test skill (if you do manual test, you can also find opportunities. But if you want to succeed in automation testing, you have to be comfortable for technical things, using different tools and different languages, CICD techniques.); attention to details;
P10	Attention to details; Analytical skill (you should both look in a bigger picture and understand details); technical skills (not just automation skills, but also knowledge in continuous deployment and devops); Communication skills: collaborate ; Resourceful: E.g. when you find a stack overflow, you cannot always find developers to solve the problems. you have to learn on you own when you need it. the reality is when you	pay atttention to both small details and big picture. You should keep your eye on the whole conception. Not just bug testing, and breaking the application. Analytical skills: put yourself as a "real" user, think outside of the box.

Table E.2: Information Collected from Interview Questions Q14-16



Interviewee No	Junior	Intermediate	Senior	Lead	Manager
P1	IT background; passion in testing;	using different test tools, techniques, programming languages; different way of thinking. Have a big picture first.	Intermediate are stilling developing their own view and way of doing things. But Senior have different opinions on different testing areas and can learn new things quickly. Great trouble-shooting and problem-solving skill even in unfamiliar area.	Communication skills, not to do too much detail things. Make plans and strategies; Arrange everything; High level knowledge/view (Such as what tools to use)	Lead and Manager are different in different companies
P2	can execute test cases, but not good at analyzing problems independently	can finish test tasks independently (not the whole test project), begin to think critically and independently and can summarize the prior experience.	Can guide the whole project, and think globally. Have overall view of the project. Can mentor and coach juniors. Most focus on technical things.	always the senior tester who want to move up to TM. Have more skills in management and coordination.	have great communication skills and coordination skills; can be responsible for coordinate multi-project, familiar with business domain, but not necessary technical knowledge
P3	execute test case;	Design test cases; Can finish some tasks independently.	Focus on Strategy, risk assessment and quality control; Can mentor juniors; Mostly focus on technical things. Almost any Senior tester can be a Test Lead.	more responsibility in management than Senior tester; More focus on a simple project, including its strategy	More focus on the development direction of the whole department, is responsible for the policy, and quality criteria.
P4	Provide them a list to check; Can explore on their own and do things as well, but you just provide them a list to check. Junior works with the whole team, while intermediate work on their own.	The intermediate tester know the direction of how to finish the tasks. The scenario is: The senior tester or lead say: hey, go to look at this area. And the intermediate tester will come back and report: I have found this kind of things. Intermediate tester is kind of in the level of direction and guidance. Give you a big direction and you can finish it by yourself. They can work on their own for specific testing task.	Senior testers focus on how to manage tasks or workload. If they need help, they will go and ask for help by themselves. Be more proactive, and can make judgment by themselves Senior: mostly responsible for themselves and their team to deliver project, more technical thing. (project management)	for people, manage people, and how other people around are doing (team management)	Don't have manager. Tech Lead (lead of the team, maybe senior developer in the team): responsible for the project; Feature Lead: when we have work on a feature, they look after the delivery of the features; QA coach: help skills need for the work; like how are you? How you work with others; work together more smoothly. QA lead: how the quality is achieved. QA goal.
P5	be told what to do	they can do a certain thing from start to finish: finish some task independently. know test strategy	They have to teach/mentor other people, and they should come up with creative way of how to test things; They know you cannot test everything and knowing what to test. They have good judgement and make decision of what is enough. (make decision on what to test, how to test and why?); Test strategy is just evaluating the risk and what should be test first (priority).	lead the team to a better situation. help and grow the skills of fellow tester, rather than dictating tester to do things. Allow others to try things on their own.	managers is to manage other people, quite like the things of process and policy. They care about the project delivery.
P6	mind holding	work independently; test design; analyze business logic; communication skill; track defects;	mentor junior; bring new idea to the company; leadership;		strategies in high level; champion company; improvement; interviewing; write job adverts; performance objectives; align with the company policy; business, such as training testers, commercially aware;

Table E.2: Information Collected from Interview Questions Q14-16



Interviewee No	Junior	Intermediate	Senior	Lead	Manager
P6	mind holding	work independently; test design; analyze business logic; communication skill; track defects;	mentor junior; bring new idea to the company; leadership;		strategies in high level; champion company; improvement; interviewing; write job adverts; performance objectives; align with the company policy; business, such as training testers, commercially aware;
P7		be agile; learn from developer and yourself; have better relationships with developer; learn more knowledge from testing books; verifying if you are doing the right thing (go to meetup); ask questions; check test pyramid (some test are in the wrong level. e.g. developer don't do much	experience in where to find problems and where to put effort;		
P8	Junior may have not done test before. Have basic technical knowledge, or IT background, such as code or sql; concepts (tell you one concept, you can fill it up).	some level coding; tools experience and knowledge on workflow; planning strategies; ISTQB or standard;	know the product, and have boarder scope of what you are looking at (looking at toolset, what to use for end test, which test level are the tools related to, and where to apply); test strategies and planning (you can't do it with less requirement); Coding background of automation; as no huge tones of documentation, there is gap between governance; know what to test or not in head (what happens here, it should be tested);	leadership; set goals or paths; decide test tool sets; manage and review workflow; protect team (let the team work);	resource person
P9	more of executing the scenarios (look at the cases and execute them); explain to them how to use the tool;	can write and report them; talk to stakeholders; analysis of results; the tools to use; represent the team in the scrum meeting;	teach or mentor the junior testers; review work of junior and intermediate testers, when changing some code; represent the team in different situations, including speak in the scrum meeting, talking to clients;		less use for TM. Scrum master take the role, such as talking to different teams
P10	test scenarios; communication skills (speak up and not afraid to ask questions be aware you are part of a team, quality is not just with the tester.);	impact analysis; have more experience and knowledge in different domain than junior (e.g. you know not just the UI but also what's happening in the back);	make decisions on different things such as the scope of project; technical skills: such as automation test (Not just coding, but also analyzing and validating);assign or delegate tasks to other people; mentoring or trainning others e.g. tools; give feedback to stakeholders in the whole view; guidance in how to solve things on your experience; have value idea for the programme.	make decisions and estimate impacts.	

Appendix F

Importance of Skills and Capabilities

F.1 Frequency of Skills and Capabilities in the Sample of Job Adverts

Table F.1: Frequency of Skills and Capabilities in the Sample of Job Adverts

Rank	Skills and Capabilities	Frequency	Percentage
1	Ability to do automation testing including scripting	73	62%
2	Ability to use appropriate testing tools and frameworks	64	55%
3	General communication skills	63	54%
4	Software development process knowledge	55	47%
5	Programming language knowledge	52	44%
6	Problem solving	51	44%
7	Business domain knowledge	39	33%
8	Analytical skills	37	32%
8	Web domain knowledge	37	32%
10	Ability to work with people at all levels of an organisation	33	28%
11	Team player	32	27%
12	Database knowledge	26	22%
13	Ability to create and run acceptance tests	24	21%
13	Ability to write test plans	24	21%
15	Leadership	23	20%
16	Ability to create and run integration testing	22	19%
16	QA skills other than testing	22	19%
16	Ability to write test cases	22	19%
16	Ability to write useful testing documentation	22	19%
20	Ability to build relationships	21	18%
20	Cloud and network infrastructure knowledge	21	18%
22	Ability to do non-functional testing	20	17%
23	Oral presentation skills	18	15%
24	Writing skills	17	15%
24	Autonomous	17	15%
26	Passionate	16	14%
27	Adaptable	15	13%
27	Mobile domain knowledge	15	13%
29	Proactive	13	11%
29	Attention to details	13	11%
29	Operation System knowledge	13	11%
32	Positive attitude	12	10%
33	Ability to think strategically	9	8%
34	Trouble shooting /debugging	8	7%
34	Creative	8	7%
34	Self-motivated	8	7%
34	Firmware domain knowledge	8	7%

Table F.1: Frequency of Skills and Capabilities in the Sample of Job Adverts

Rank	Skills and Capabilities	Frequency	Percentage
38	Enjoying a challenge	7	6%
39	Customer oriented	6	5%
39	Willing to learn/share	6	5%
39	Pragmatic	6	5%
39	Responsible	6	5%
43	Good judgement	5	4%
43	Ability to use Test Driven Development (TDD)/ Behaviour Driven Development (BDD)	5	4%
45	Good sense of humour	4	3%
45	Ability to work under pressure	4	3%
45	Curious	4	3%
45	Energetic	4	3%
49	Good language/English skill	3	3%
49	Dedicated	3	3%
51	Active listener	2	2%
51	Ambitious	2	2%
53	Hard working	1	1%

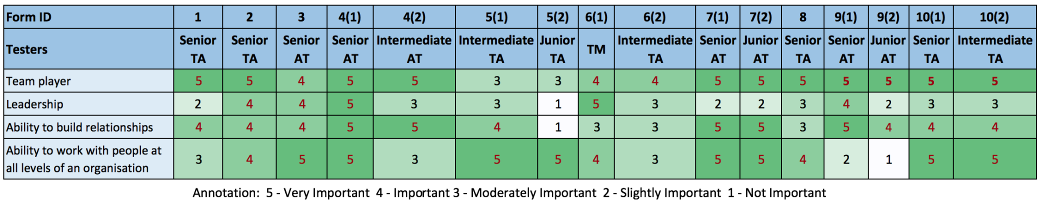
F.2 The Importance Levels of Likert Scale Obtained from Interviews

Table F.2: The Importance Levels of Communication Skills Obtained from Interviews

Form ID	1	2	3	4(1)	4(2)	5(1)	5(2)	6(1)	6(2)	7(1)	7(2)	8	9(1)	9(2)	10(1)	10(2)
Testers	Senior TA	Senior TA	Senior AT	Senior AT	Intermediate AT	Intermediate TA	Junior TA	TM	Intermediate TA	Senior AT	Junior AT	Senior TA	Senior AT	Junior AT	Senior TA	Intermediate TA
Oral presentation skills	5	5	4	4	3	4	2	4	3	2	2	2	5	3	3	3
Writing skills	2	4	5	5	5	4	2	4	3	1	1	4	4	4	2	2
Persuasive	4	4	4	2	4	3	2	4	3	1	1	3	4	3	5	4
Active listener	4	5	4	5	5	4	4	4	3	5	5	4	5	5	5	5
Empathetic to others	3	3.5	4	4	4	3.5	2	3	3	4	4	2	5	5	4	4
Good language/English skill	3.5	5	4	4	4	4	3	4	3	3	3	3	5	5	5	5

Annotation: 5 - Very Important 4 - Important 3 - Moderately Important 2 - Slightly Important 1 - Not Important

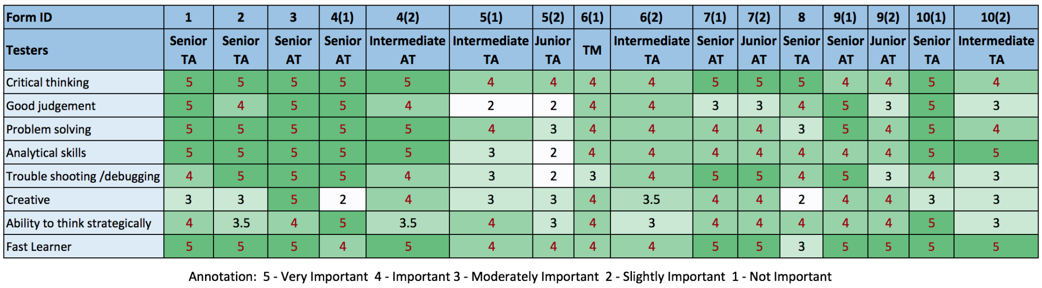
Table F.3: The Importance Levels of Social Skills Obtained from Interviews



Form ID	1	2	3	4(1)	4(2)	5(1)	5(2)	6(1)	6(2)	7(1)	7(2)	8	9(1)	9(2)	10(1)	10(2)
Testers	Senior TA	Senior TA	Senior AT	Senior AT	Intermediate AT	Intermediate TA	Junior TA	TM	Intermediate TA	Senior AT	Junior AT	Senior TA	Senior AT	Junior AT	Senior TA	Intermediate TA
Team player	5	5	4	5	5	3	3	4	4	5	5	5	5	5	5	5
Leadership	2	4	4	5	3	3	1	5	3	2	2	3	4	2	3	3
Ability to build relationships	4	4	4	5	5	4	1	3	3	5	5	3	5	4	4	4
Ability to work with people at all levels of an organisation	3	4	5	5	3	5	5	4	3	5	5	4	2	1	5	5

Annotation: 5 - Very Important 4 - Important 3 - Moderately Important 2 - Slightly Important 1 - Not Important

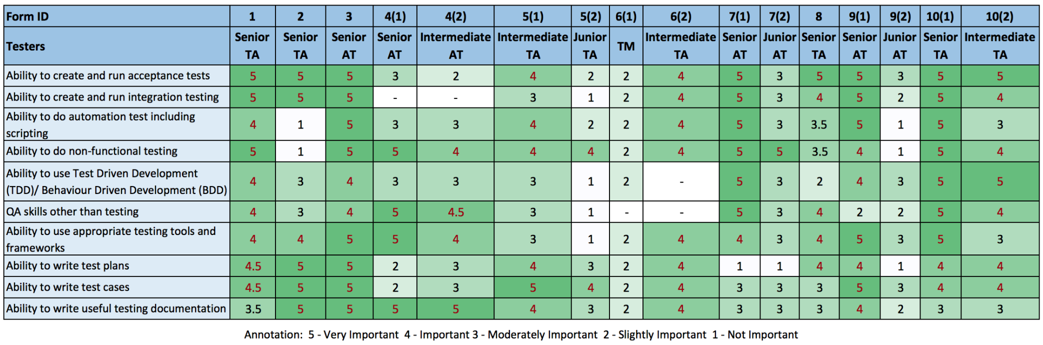
Table F.4: The Importance Levels of Cognitive Skills Obtained from Interviews



Form ID	1	2	3	4(1)	4(2)	5(1)	5(2)	6(1)	6(2)	7(1)	7(2)	8	9(1)	9(2)	10(1)	10(2)
Testers	Senior TA	Senior TA	Senior AT	Senior AT	Intermediate AT	Intermediate TA	Junior TA	TM	Intermediate TA	Senior AT	Junior AT	Senior TA	Senior AT	Junior AT	Senior TA	Intermediate TA
Critical thinking	5	5	5	5	5	4	4	4	4	5	5	5	4	4	5	4
Good judgement	5	4	5	5	4	2	2	4	4	3	3	4	5	3	5	3
Problem solving	5	5	5	5	5	4	3	4	4	4	4	3	5	4	5	4
Analytical skills	5	5	5	5	5	3	2	4	4	4	4	4	4	4	5	5
Trouble shooting /debugging	4	5	5	5	4	3	2	3	4	5	5	4	5	3	4	3
Creative	3	3	5	2	4	3	3	4	3.5	4	4	2	4	4	3	3
Ability to think strategically	4	3.5	4	5	3.5	4	3	4	3	4	4	4	4	4	5	3
Fast Learner	5	5	5	4	5	4	4	4	4	5	5	3	5	5	5	5

Annotation: 5 - Very Important 4 - Important 3 - Moderately Important 2 - Slightly Important 1 - Not Important

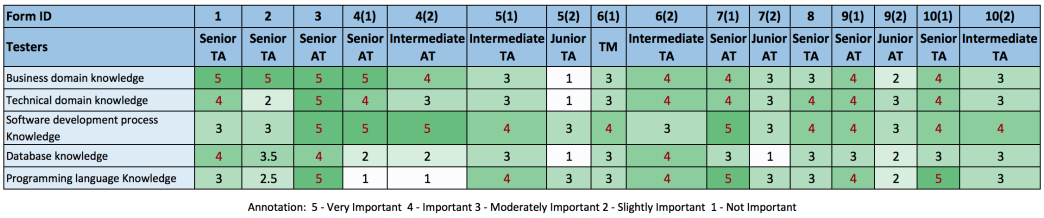
Table F.5: The Importance Levels of Testing Skills Obtained from Interviews



Form ID	1	2	3	4(1)	4(2)	5(1)	5(2)	6(1)	6(2)	7(1)	7(2)	8	9(1)	9(2)	10(1)	10(2)
Testers	Senior TA	Senior TA	Senior AT	Senior AT	Intermediate AT	Intermediate TA	Junior TA	TM	Intermediate TA	Senior AT	Junior AT	Senior TA	Senior AT	Junior AT	Senior TA	Intermediate TA
Ability to create and run acceptance tests	5	5	5	3	2	4	2	2	4	5	3	5	5	3	5	5
Ability to create and run integration testing	5	5	5	-	-	3	1	2	4	5	3	4	5	2	5	4
Ability to do automation test including scripting	4	1	5	3	3	4	2	2	4	5	3	3.5	5	1	5	3
Ability to do non-functional testing	5	1	5	5	4	4	4	2	4	5	5	3.5	4	1	5	4
Ability to use Test Driven Development (TDD)/ Behaviour Driven Development (BDD)	4	3	4	3	3	3	1	2	-	5	3	2	4	3	5	5
QA skills other than testing	4	3	4	5	4.5	3	1	-	-	5	3	4	2	2	5	4
Ability to use appropriate testing tools and frameworks	4	4	5	5	4	3	1	2	4	4	3	4	5	3	5	3
Ability to write test plans	4.5	5	5	2	3	4	3	2	4	1	1	4	4	1	4	4
Ability to write test cases	4.5	5	5	2	3	5	4	2	4	3	3	3	5	3	4	4
Ability to write useful testing documentation	3.5	5	5	5	5	4	3	2	4	3	3	3	4	2	3	3

Annotation: 5 - Very Important 4 - Important 3 - Moderately Important 2 - Slightly Important 1 - Not Important

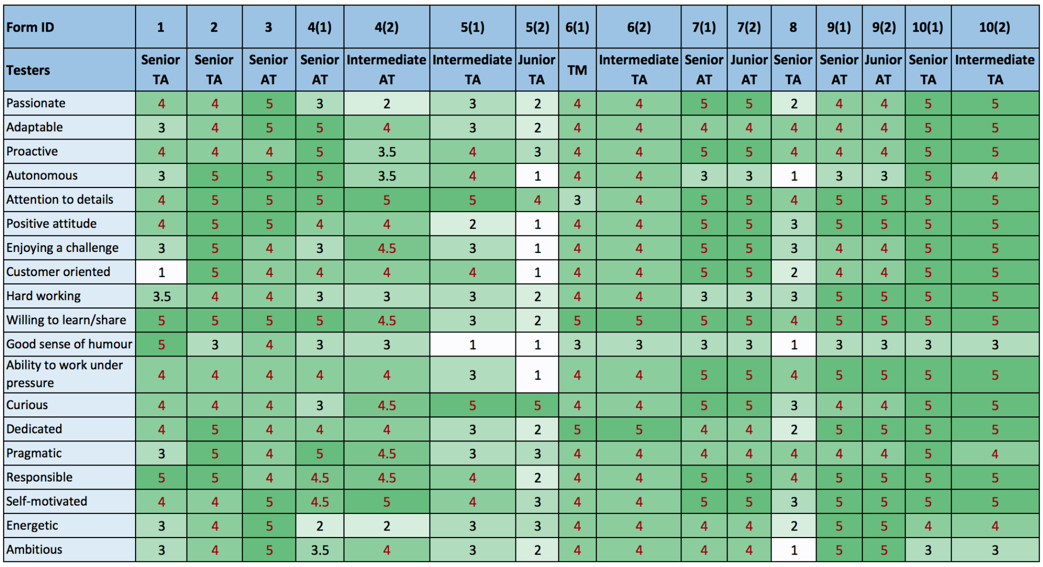
Table F.6: The Importance Levels of Other Knowledge Obtained from Interviews



Form ID	1	2	3	4(1)	4(2)	5(1)	5(2)	6(1)	6(2)	7(1)	7(2)	8	9(1)	9(2)	10(1)	10(2)
Testers	Senior TA	Senior TA	Senior AT	Senior AT	Intermediate AT	Intermediate TA	Junior TA	TM	Intermediate TA	Senior AT	Junior AT	Senior TA	Senior AT	Junior AT	Senior TA	Intermediate TA
Business domain knowledge	5	5	5	5	4	3	1	3	4	4	3	3	4	2	4	3
Technical domain knowledge	4	2	5	4	3	3	1	3	4	4	3	4	4	3	4	3
Software development process Knowledge	3	3	5	5	5	4	3	4	3	5	3	4	4	3	4	4
Database knowledge	4	3.5	4	2	2	3	1	3	4	3	1	3	3	2	3	3
Programming language Knowledge	3	2.5	5	1	1	4	3	3	4	5	3	3	4	2	5	3

Annotation: 5 - Very Important 4 - Important 3 - Moderately Important 2 - Slightly Important 1 - Not Important

Table F.7: The Importance Levels of Personal Qualities Obtained from Interviews



Form ID	1	2	3	4(1)	4(2)	5(1)	5(2)	6(1)	6(2)	7(1)	7(2)	8	9(1)	9(2)	10(1)	10(2)
Testers	Senior TA	Senior TA	Senior AT	Senior AT	Intermediate AT	Intermediate TA	Junior TA	TM	Intermediate TA	Senior AT	Junior AT	Senior TA	Senior AT	Junior AT	Senior TA	Intermediate TA
Passionate	4	4	5	3	2	3	2	4	4	5	5	2	4	4	5	5
Adaptable	3	4	5	5	4	3	2	4	4	4	4	4	4	4	5	5
Proactive	4	4	4	5	3.5	4	3	4	4	5	5	4	4	4	5	5
Autonomous	3	5	5	5	3.5	4	1	4	4	3	3	1	3	3	5	4
Attention to details	4	5	5	5	5	5	4	3	4	5	5	4	5	5	5	5
Positive attitude	4	5	5	4	4	2	1	4	4	5	5	3	5	5	5	5
Enjoying a challenge	3	5	4	3	4.5	3	1	4	4	5	5	3	4	4	5	5
Customer oriented	1	5	4	4	4	4	1	4	4	5	5	2	4	4	5	5
Hard working	3.5	4	4	3	3	3	2	4	4	3	3	3	5	5	5	5
Willing to learn/share	5	5	5	5	4.5	3	2	5	5	5	5	4	5	5	5	5
Good sense of humour	5	3	4	3	3	1	1	3	3	3	3	1	3	3	3	3
Ability to work under pressure	4	4	4	4	4	3	1	4	4	5	5	4	5	5	5	5
Curious	4	4	4	3	4.5	5	5	4	4	5	5	3	4	4	5	5
Dedicated	4	5	4	4	4	3	2	5	5	4	4	2	5	5	5	5
Pragmatic	3	5	4	5	4.5	3	3	4	4	4	4	4	4	4	5	4
Responsible	5	5	4	4.5	4.5	4	2	4	4	5	5	4	5	5	5	5
Self-motivated	4	4	5	4.5	5	4	3	4	4	5	5	3	5	5	5	5
Energetic	3	4	5	2	2	3	3	4	4	4	4	2	5	5	4	4
Ambitious	3	4	5	3.5	4	3	2	4	4	4	4	1	5	5	3	3

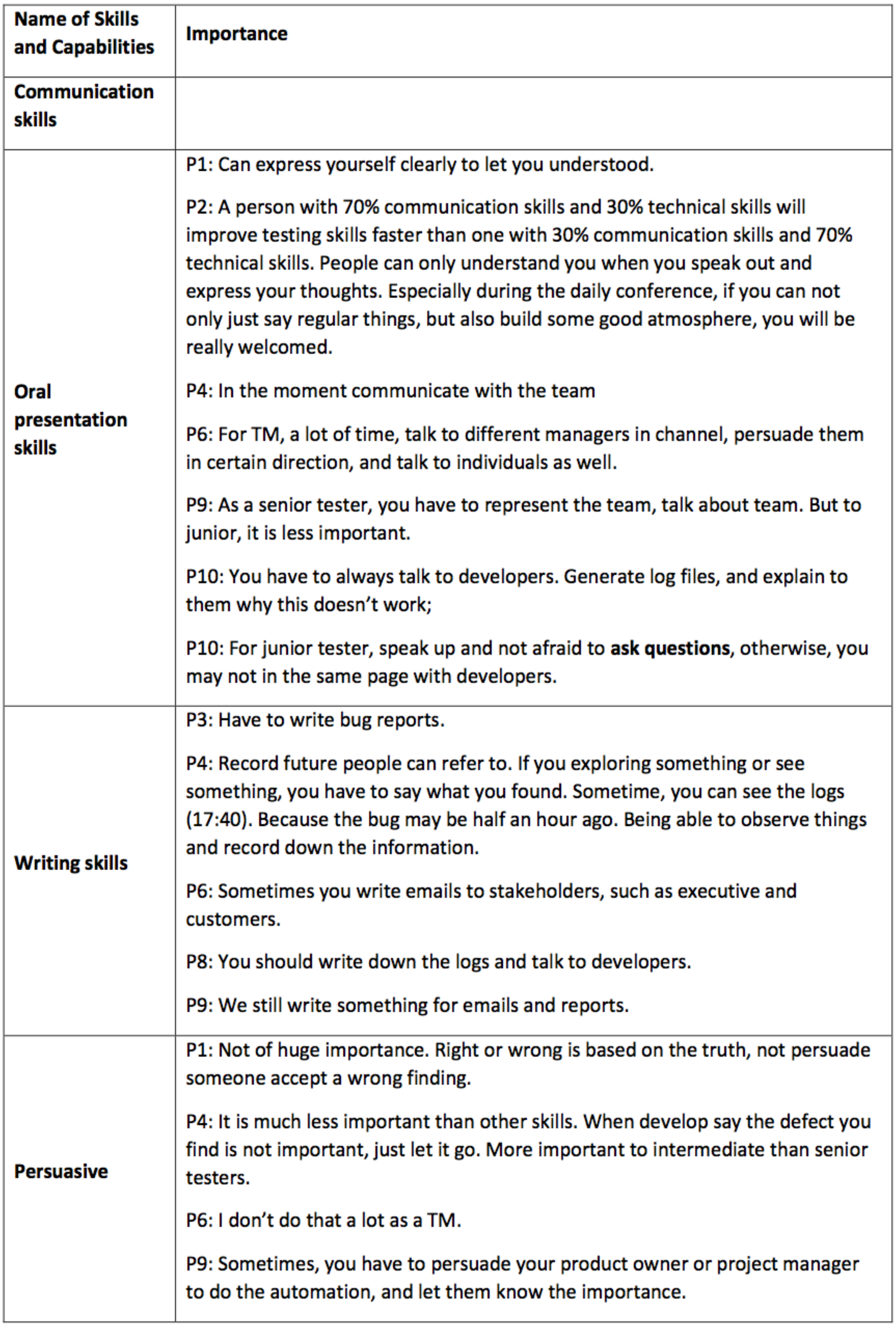
Annotation: 5 - Very Important 4 - Important 3 - Moderately Important 2 - Slightly Important 1 - Not Important

Appendix G

The Reasons for the Importance Levels of Skills and Capabilities

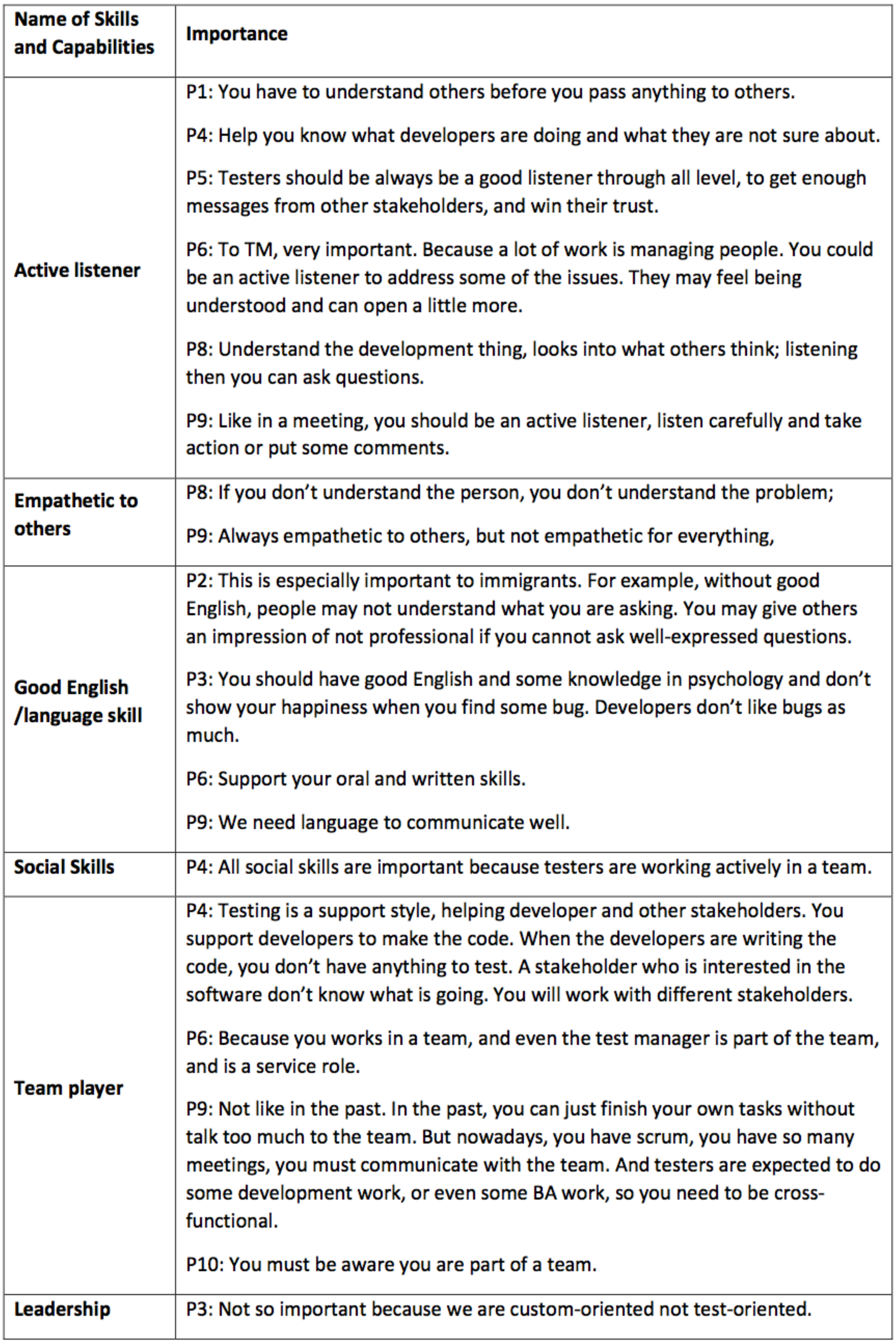
Annotation: P is the abbreviation of participant; P1, P2, ..., and P10 denote the number given to interviewees.

Table G.1: Reasons for Importance Levels



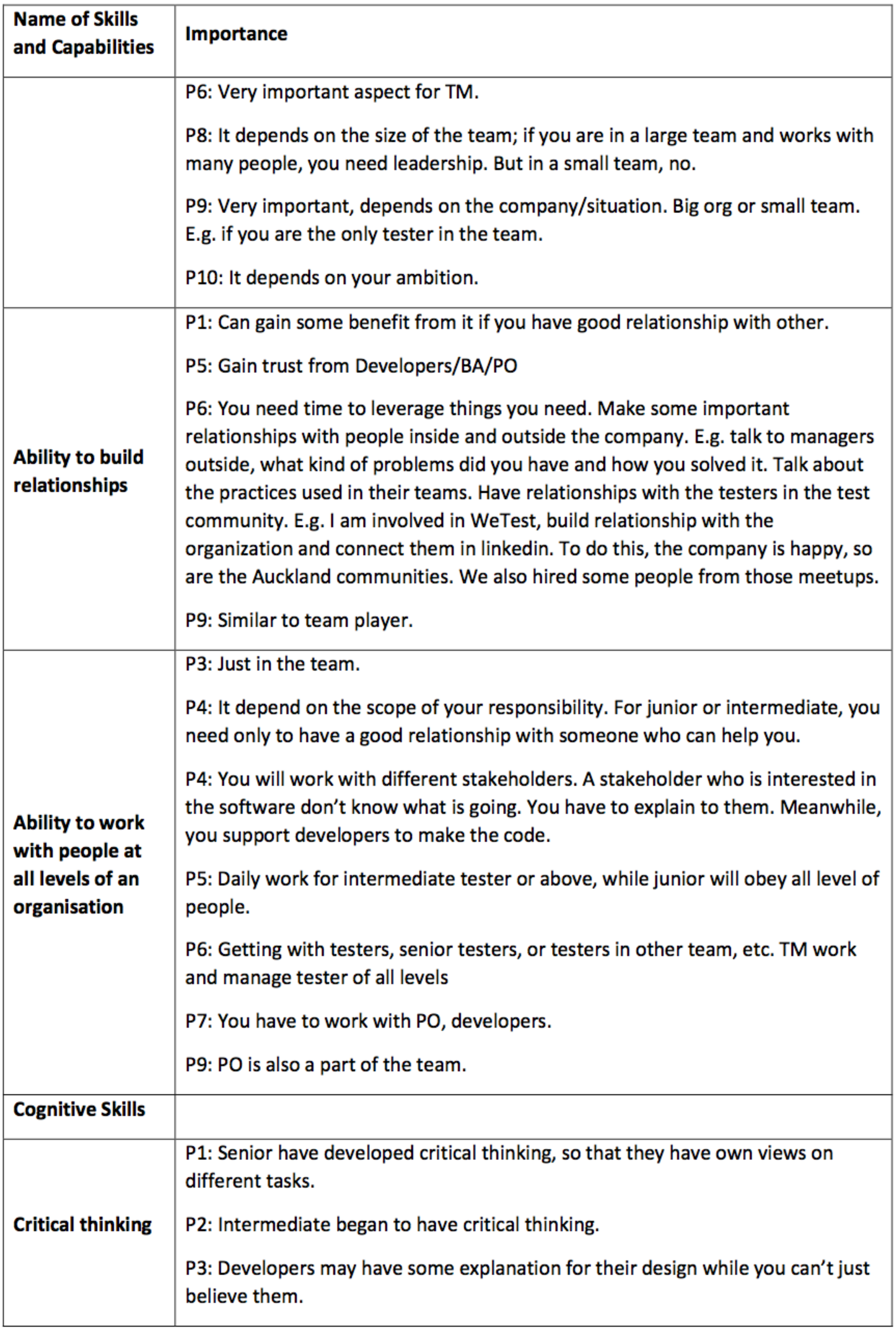
Name of Skills and Capabilities	**Importance**
Communication skills	
Oral presentation skills	P1: Can express yourself clearly to let you understood. P2: A person with 70% communication skills and 30% technical skills will improve testing skills faster than one with 30% communication skills and 70% technical skills. People can only understand you when you speak out and express your thoughts. Especially during the daily conference, if you can not only just say regular things, but also build some good atmosphere, you will be really welcomed. P4: In the moment communicate with the team P6: For TM, a lot of time, talk to different managers in channel, persuade them in certain direction, and talk to individuals as well. P9: As a senior tester, you have to represent the team, talk about team. But to junior, it is less important. P10: You have to always talk to developers. Generate log files, and explain to them why this doesn't work; P10: For junior tester, speak up and not afraid to **ask questions**, otherwise, you may not in the same page with developers.
Writing skills	P3: Have to write bug reports. P4: Record future people can refer to. If you exploring something or see something, you have to say what you found. Sometime, you can see the logs (17:40). Because the bug may be half an hour ago. Being able to observe things and record down the information. P6: Sometimes you write emails to stakeholders, such as executive and customers. P8: You should write down the logs and talk to developers. P9: We still write something for emails and reports.
Persuasive	P1: Not of huge importance. Right or wrong is based on the truth, not persuade someone accept a wrong finding. P4: It is much less important than other skills. When develop say the defect you find is not important, just let it go. More important to intermediate than senior testers. P6: I don't do that a lot as a TM. P9: Sometimes, you have to persuade your product owner or project manager to do the automation, and let them know the importance.

Table G.1: Reasons for Importance Levels



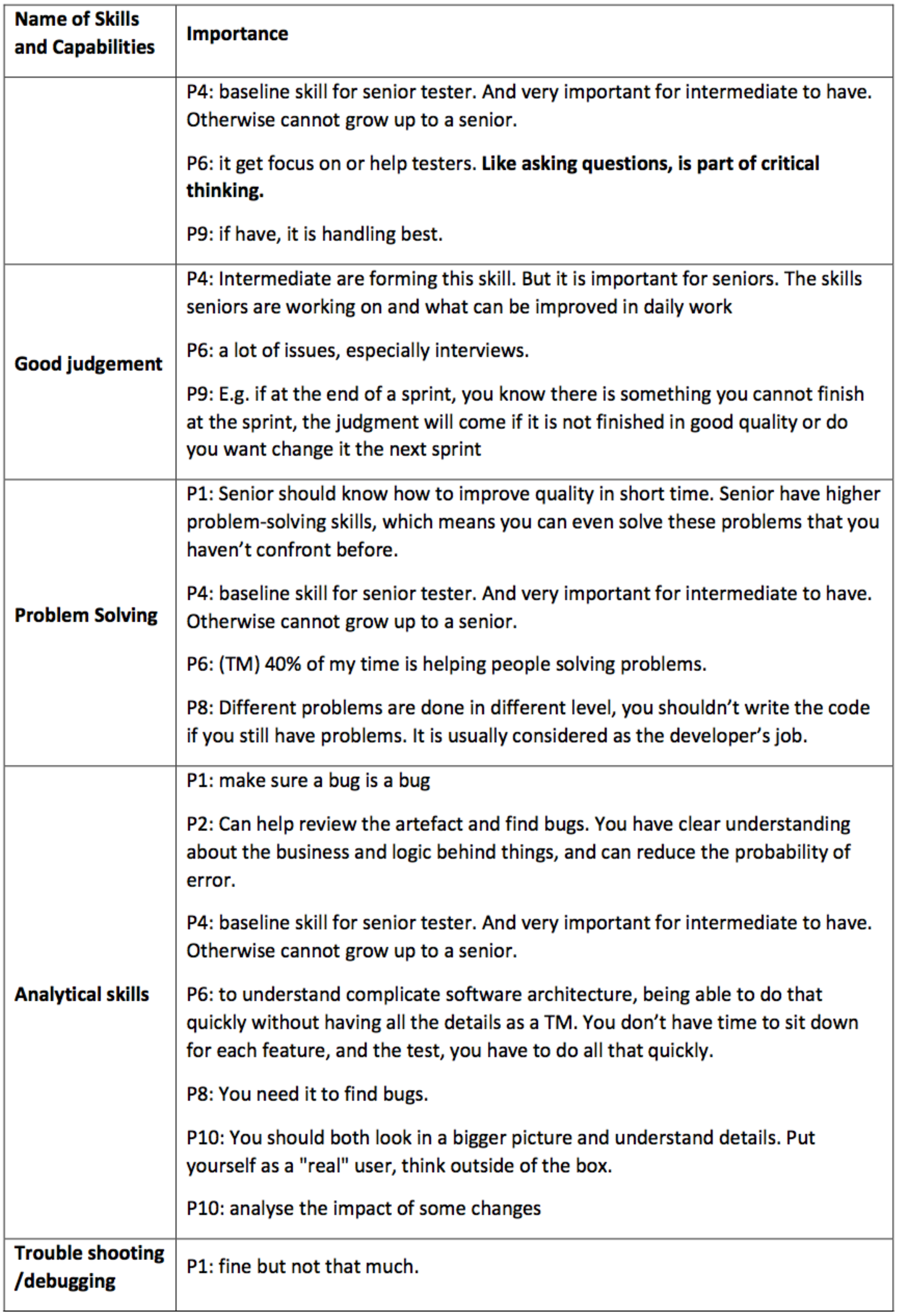
Name of Skills and Capabilities	Importance
Active listener	P1: You have to understand others before you pass anything to others. P4: Help you know what developers are doing and what they are not sure about. P5: Testers should be always be a good listener through all level, to get enough messages from other stakeholders, and win their trust. P6: To TM, very important. Because a lot of work is managing people. You could be an active listener to address some of the issues. They may feel being understood and can open a little more. P8: Understand the development thing, looks into what others think; listening then you can ask questions. P9: Like in a meeting, you should be an active listener, listen carefully and take action or put some comments.
Empathetic to others	P8: If you don't understand the person, you don't understand the problem; P9: Always empathetic to others, but not empathetic for everything,
Good English /language skill	P2: This is especially important to immigrants. For example, without good English, people may not understand what you are asking. You may give others an impression of not professional if you cannot ask well-expressed questions. P3: You should have good English and some knowledge in psychology and don't show your happiness when you find some bug. Developers don't like bugs as much. P6: Support your oral and written skills. P9: We need language to communicate well.
Social Skills	P4: All social skills are important because testers are working actively in a team.
Team player	P4: Testing is a support style, helping developer and other stakeholders. You support developers to make the code. When the developers are writing the code, you don't have anything to test. A stakeholder who is interested in the software don't know what is going. You will work with different stakeholders. P6: Because you works in a team, and even the test manager is part of the team, and is a service role. P9: Not like in the past. In the past, you can just finish your own tasks without talk too much to the team. But nowadays, you have scrum, you have so many meetings, you must communicate with the team. And testers are expected to do some development work, or even some BA work, so you need to be cross-functional. P10: You must be aware you are part of a team.
Leadership	P3: Not so important because we are custom-oriented not test-oriented.

Table G.1: Reasons for Importance Levels



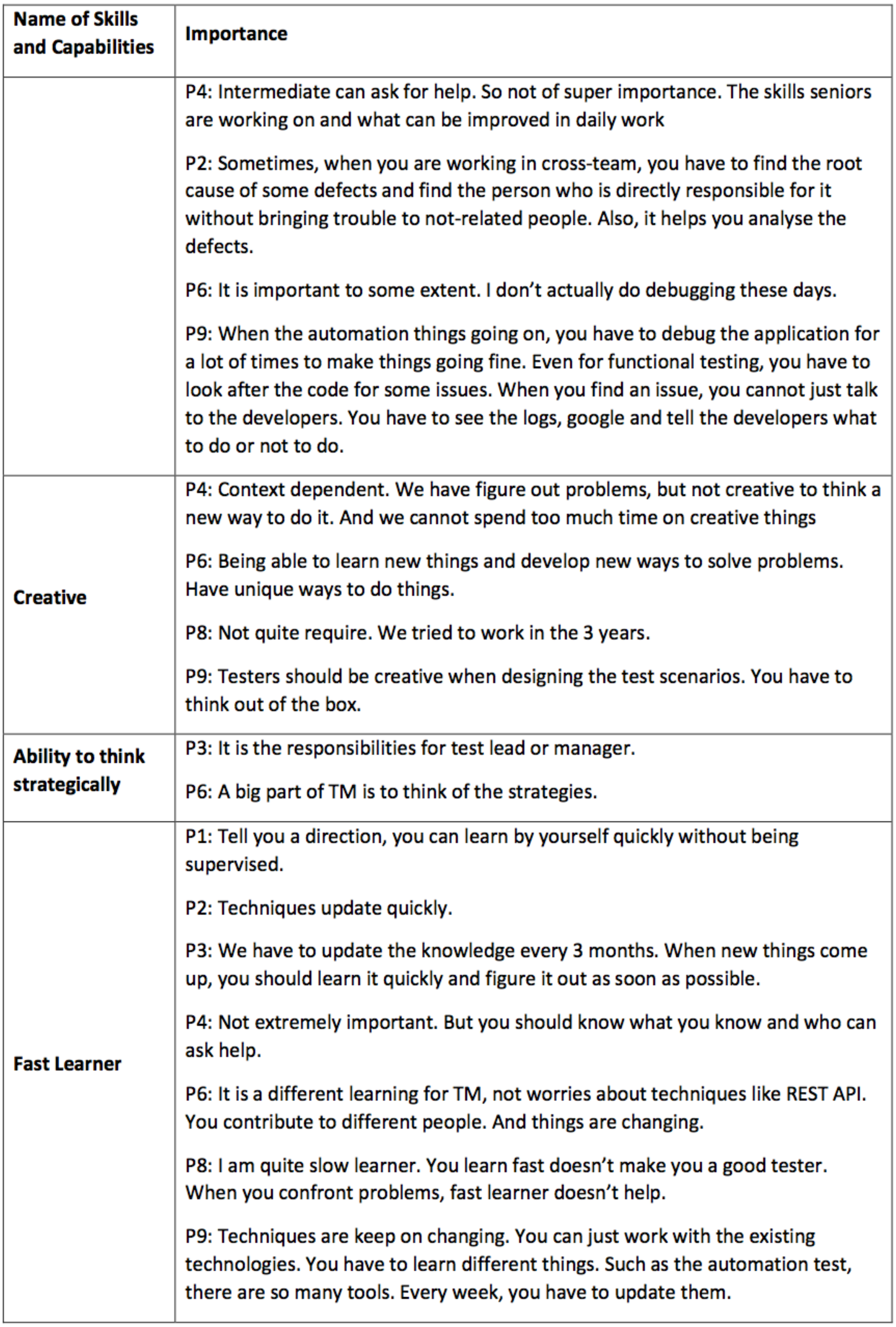
Name of Skills and Capabilities	**Importance**
	P6: Very important aspect for TM. P8: It depends on the size of the team; if you are in a large team and works with many people, you need leadership. But in a small team, no. P9: Very important, depends on the company/situation. Big org or small team. E.g. if you are the only tester in the team. P10: It depends on your ambition.
Ability to build relationships	P1: Can gain some benefit from it if you have good relationship with other. P5: Gain trust from Developers/BA/PO P6: You need time to leverage things you need. Make some important relationships with people inside and outside the company. E.g. talk to managers outside, what kind of problems did you have and how you solved it. Talk about the practices used in their teams. Have relationships with the testers in the test community. E.g. I am involved in WeTest, build relationship with the organization and connect them in linkedin. To do this, the company is happy, so are the Auckland communities. We also hired some people from those meetups. P9: Similar to team player.
Ability to work with people at all levels of an organisation	P3: Just in the team. P4: It depend on the scope of your responsibility. For junior or intermediate, you need only to have a good relationship with someone who can help you. P4: You will work with different stakeholders. A stakeholder who is interested in the software don't know what is going. You have to explain to them. Meanwhile, you support developers to make the code. P5: Daily work for intermediate tester or above, while junior will obey all level of people. P6: Getting with testers, senior testers, or testers in other team, etc. TM work and manage tester of all levels P7: You have to work with PO, developers. P9: PO is also a part of the team.
Cognitive Skills	
Critical thinking	P1: Senior have developed critical thinking, so that they have own views on different tasks. P2: Intermediate began to have critical thinking. P3: Developers may have some explanation for their design while you can't just believe them.

Table G.1: Reasons for Importance Levels



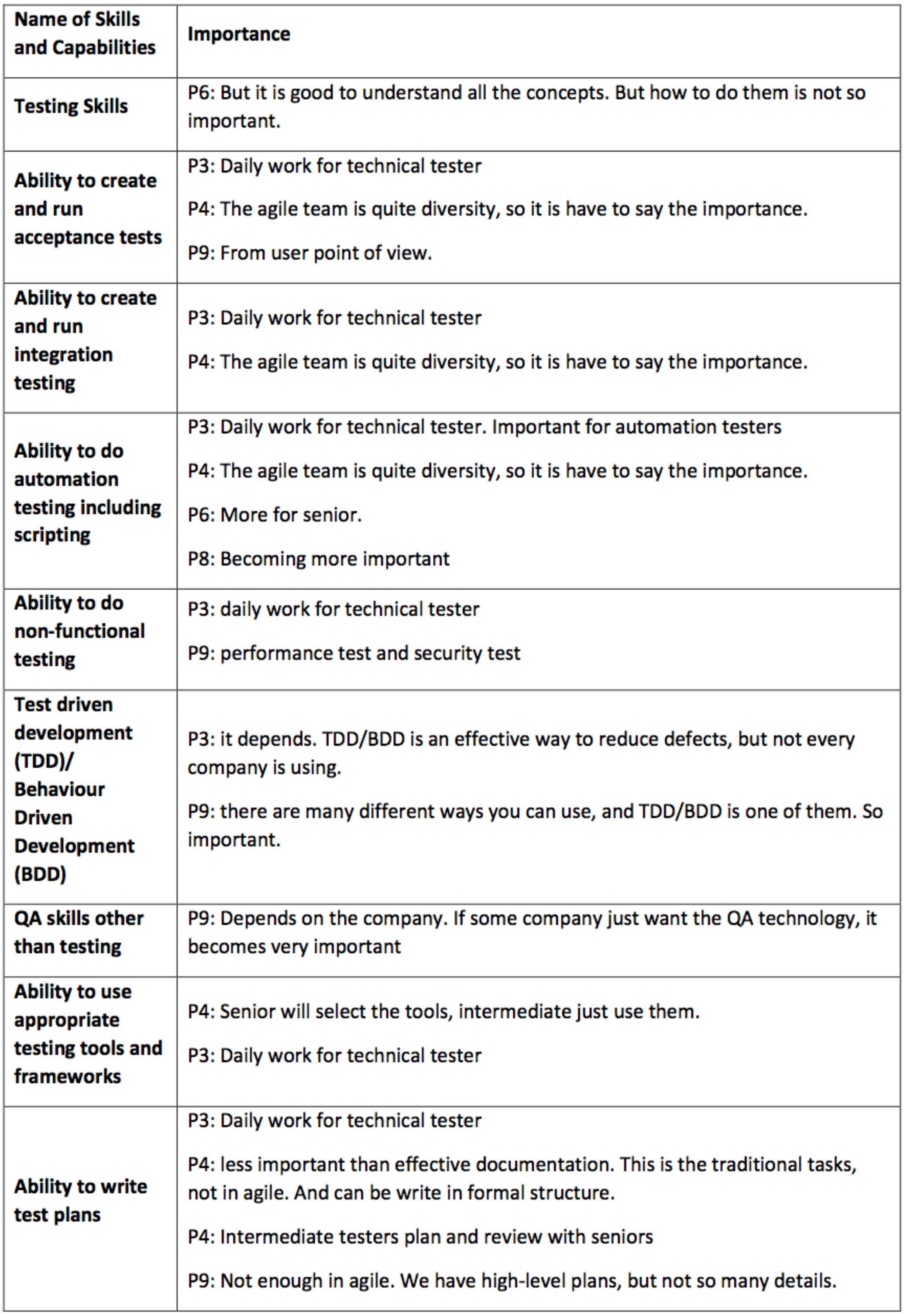
Name of Skills and Capabilities	Importance
	P4: baseline skill for senior tester. And very important for intermediate to have. Otherwise cannot grow up to a senior. P6: it get focus on or help testers. **Like asking questions, is part of critical thinking.** P9: if have, it is handling best.
Good judgement	P4: Intermediate are forming this skill. But it is important for seniors. The skills seniors are working on and what can be improved in daily work P6: a lot of issues, especially interviews. P9: E.g. if at the end of a sprint, you know there is something you cannot finish at the sprint, the judgment will come if it is not finished in good quality or do you want change it the next sprint
Problem Solving	P1: Senior should know how to improve quality in short time. Senior have higher problem-solving skills, which means you can even solve these problems that you haven't confront before. P4: baseline skill for senior tester. And very important for intermediate to have. Otherwise cannot grow up to a senior. P6: (TM) 40% of my time is helping people solving problems. P8: Different problems are done in different level, you shouldn't write the code if you still have problems. It is usually considered as the developer's job.
Analytical skills	P1: make sure a bug is a bug P2: Can help review the artefact and find bugs. You have clear understanding about the business and logic behind things, and can reduce the probability of error. P4: baseline skill for senior tester. And very important for intermediate to have. Otherwise cannot grow up to a senior. P6: to understand complicate software architecture, being able to do that quickly without having all the details as a TM. You don't have time to sit down for each feature, and the test, you have to do all that quickly. P8: You need it to find bugs. P10: You should both look in a bigger picture and understand details. Put yourself as a "real" user, think outside of the box. P10: analyse the impact of some changes
Trouble shooting /debugging	P1: fine but not that much.

Table G.1: Reasons for Importance Levels



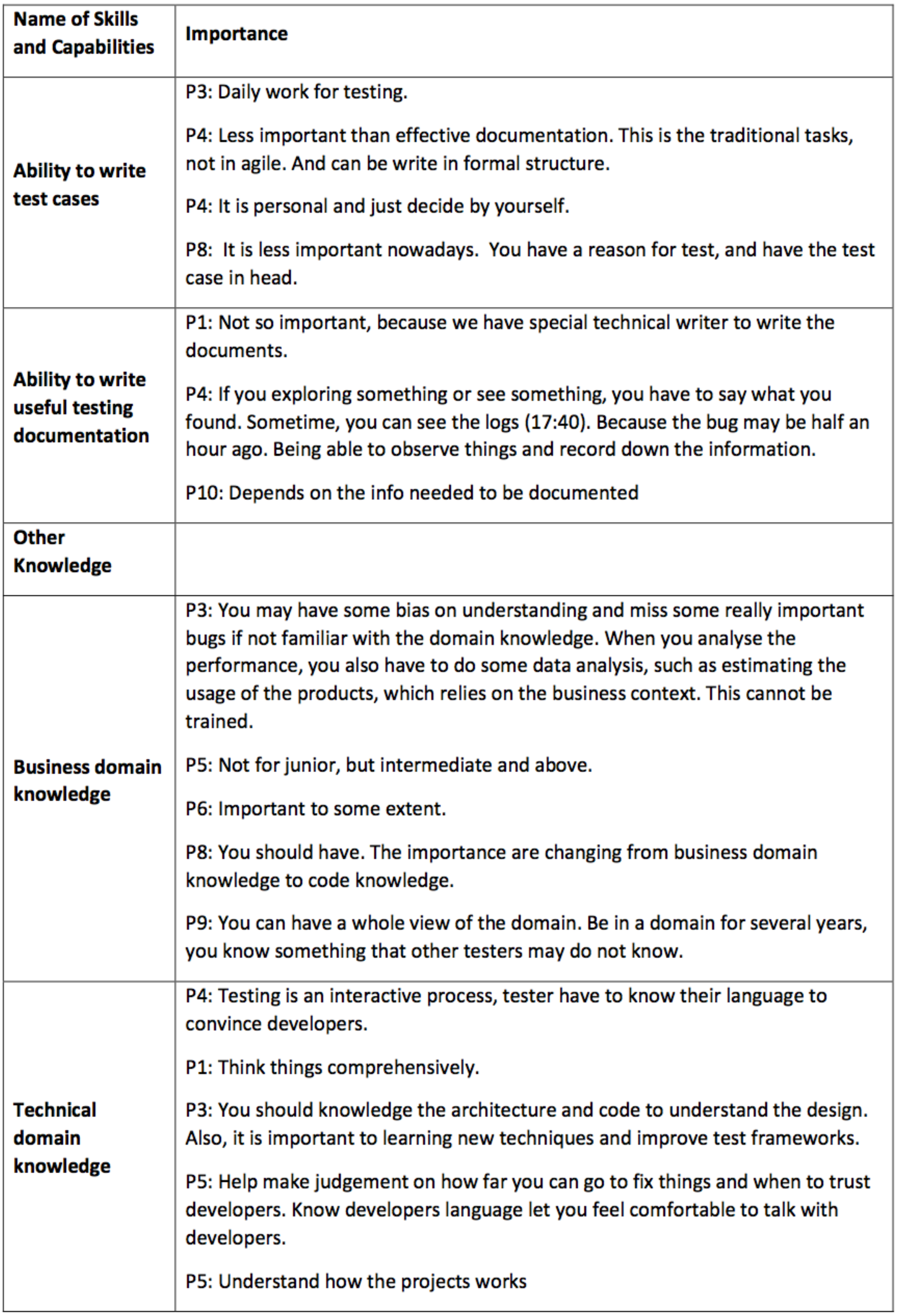
Name of Skills and Capabilities	Importance
	P4: Intermediate can ask for help. So not of super importance. The skills seniors are working on and what can be improved in daily work P2: Sometimes, when you are working in cross-team, you have to find the root cause of some defects and find the person who is directly responsible for it without bringing trouble to not-related people. Also, it helps you analyse the defects. P6: It is important to some extent. I don't actually do debugging these days. P9: When the automation things going on, you have to debug the application for a lot of times to make things going fine. Even for functional testing, you have to look after the code for some issues. When you find an issue, you cannot just talk to the developers. You have to see the logs, google and tell the developers what to do or not to do.
Creative	P4: Context dependent. We have figure out problems, but not creative to think a new way to do it. And we cannot spend too much time on creative things P6: Being able to learn new things and develop new ways to solve problems. Have unique ways to do things. P8: Not quite require. We tried to work in the 3 years. P9: Testers should be creative when designing the test scenarios. You have to think out of the box.
Ability to think strategically	P3: It is the responsibilities for test lead or manager. P6: A big part of TM is to think of the strategies.
Fast Learner	P1: Tell you a direction, you can learn by yourself quickly without being supervised. P2: Techniques update quickly. P3: We have to update the knowledge every 3 months. When new things come up, you should learn it quickly and figure it out as soon as possible. P4: Not extremely important. But you should know what you know and who can ask help. P6: It is a different learning for TM, not worries about techniques like REST API. You contribute to different people. And things are changing. P8: I am quite slow learner. You learn fast doesn't make you a good tester. When you confront problems, fast learner doesn't help. P9: Techniques are keep on changing. You can just work with the existing technologies. You have to learn different things. Such as the automation test, there are so many tools. Every week, you have to update them.

Table G.1: Reasons for Importance Levels



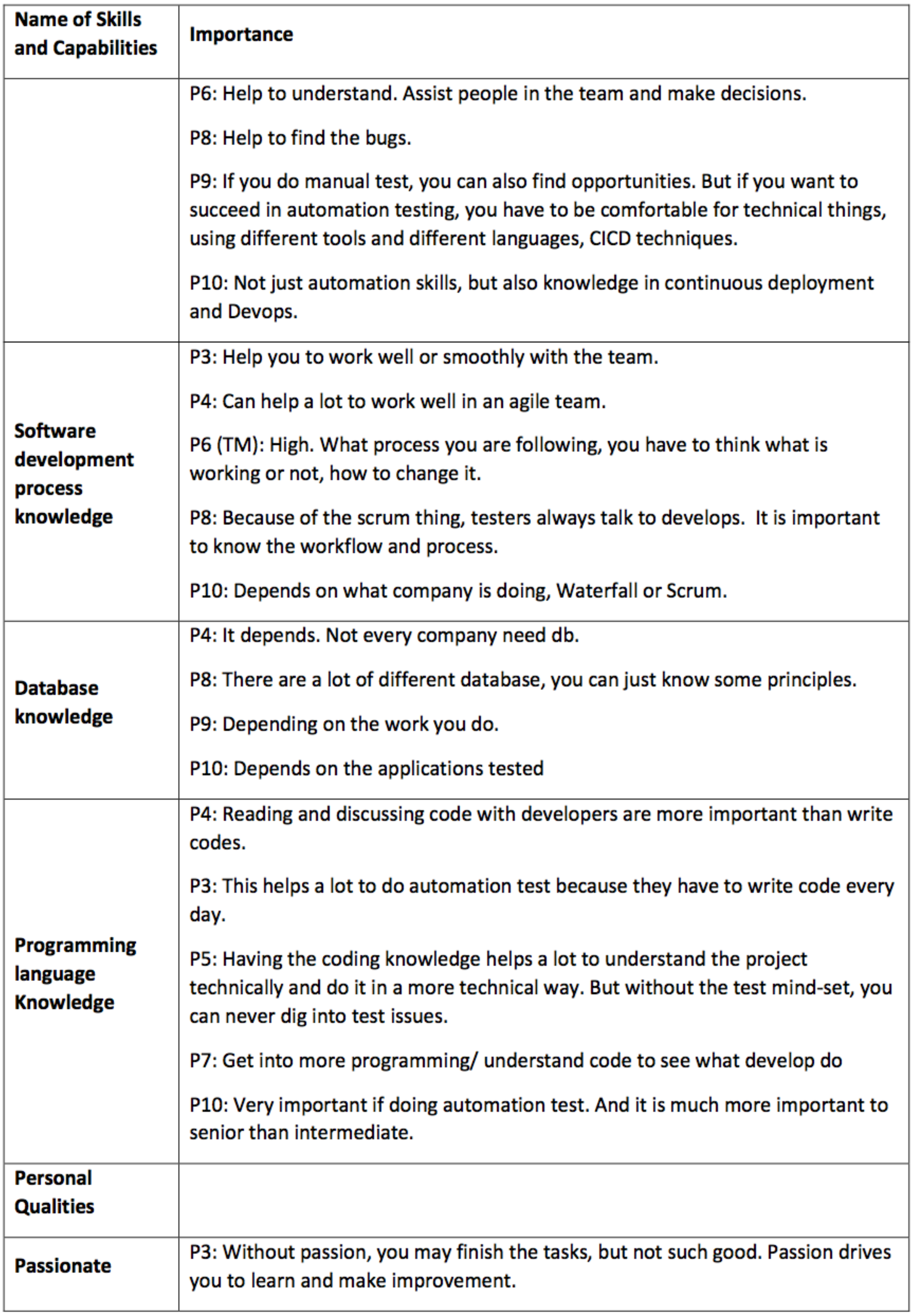
Name of Skills and Capabilities	Importance
Testing Skills	P6: But it is good to understand all the concepts. But how to do them is not so important.
Ability to create and run acceptance tests	P3: Daily work for technical tester P4: The agile team is quite diversity, so it is have to say the importance. P9: From user point of view.
Ability to create and run integration testing	P3: Daily work for technical tester P4: The agile team is quite diversity, so it is have to say the importance.
Ability to do automation testing including scripting	P3: Daily work for technical tester. Important for automation testers P4: The agile team is quite diversity, so it is have to say the importance. P6: More for senior. P8: Becoming more important
Ability to do non-functional testing	P3: daily work for technical tester P9: performance test and security test
Test driven development (TDD)/ Behaviour Driven Development (BDD)	P3: it depends. TDD/BDD is an effective way to reduce defects, but not every company is using. P9: there are many different ways you can use, and TDD/BDD is one of them. So important.
QA skills other than testing	P9: Depends on the company. If some company just want the QA technology, it becomes very important
Ability to use appropriate testing tools and frameworks	P4: Senior will select the tools, intermediate just use them. P3: Daily work for technical tester
Ability to write test plans	P3: Daily work for technical tester P4: less important than effective documentation. This is the traditional tasks, not in agile. And can be write in formal structure. P4: Intermediate testers plan and review with seniors P9: Not enough in agile. We have high-level plans, but not so many details.

Table G.1: Reasons for Importance Levels



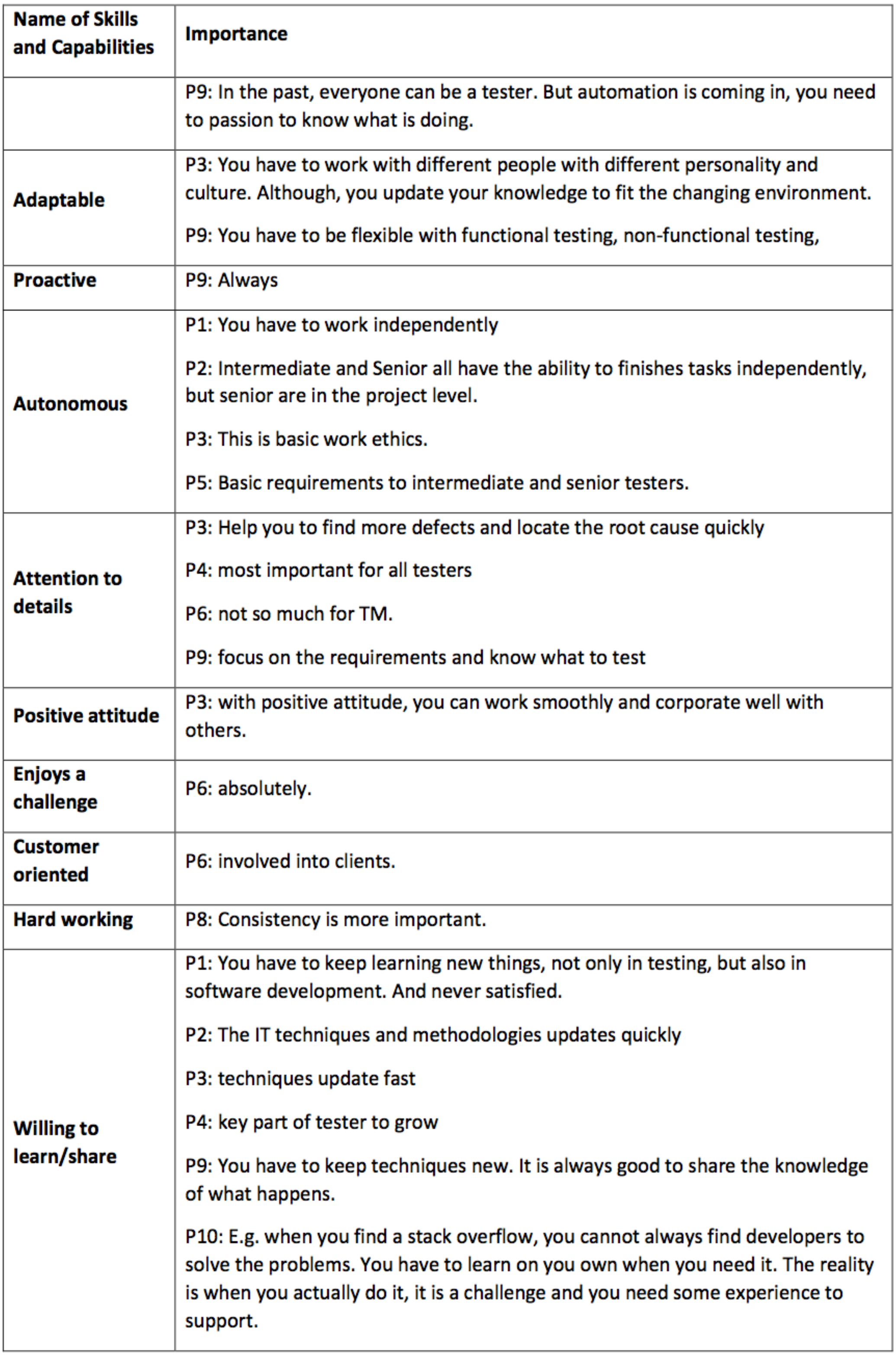
Name of Skills and Capabilities	Importance
Ability to write test cases	P3: Daily work for testing. P4: Less important than effective documentation. This is the traditional tasks, not in agile. And can be write in formal structure. P4: It is personal and just decide by yourself. P8: It is less important nowadays. You have a reason for test, and have the test case in head.
Ability to write useful testing documentation	P1: Not so important, because we have special technical writer to write the documents. P4: If you exploring something or see something, you have to say what you found. Sometime, you can see the logs (17:40). Because the bug may be half an hour ago. Being able to observe things and record down the information. P10: Depends on the info needed to be documented
Other Knowledge	
Business domain knowledge	P3: You may have some bias on understanding and miss some really important bugs if not familiar with the domain knowledge. When you analyse the performance, you also have to do some data analysis, such as estimating the usage of the products, which relies on the business context. This cannot be trained. P5: Not for junior, but intermediate and above. P6: Important to some extent. P8: You should have. The importance are changing from business domain knowledge to code knowledge. P9: You can have a whole view of the domain. Be in a domain for several years, you know something that other testers may do not know.
Technical domain knowledge	P4: Testing is an interactive process, tester have to know their language to convince developers. P1: Think things comprehensively. P3: You should knowledge the architecture and code to understand the design. Also, it is important to learning new techniques and improve test frameworks. P5: Help make judgement on how far you can go to fix things and when to trust developers. Know developers language let you feel comfortable to talk with developers. P5: Understand how the projects works

Table G.1: Reasons for Importance Levels



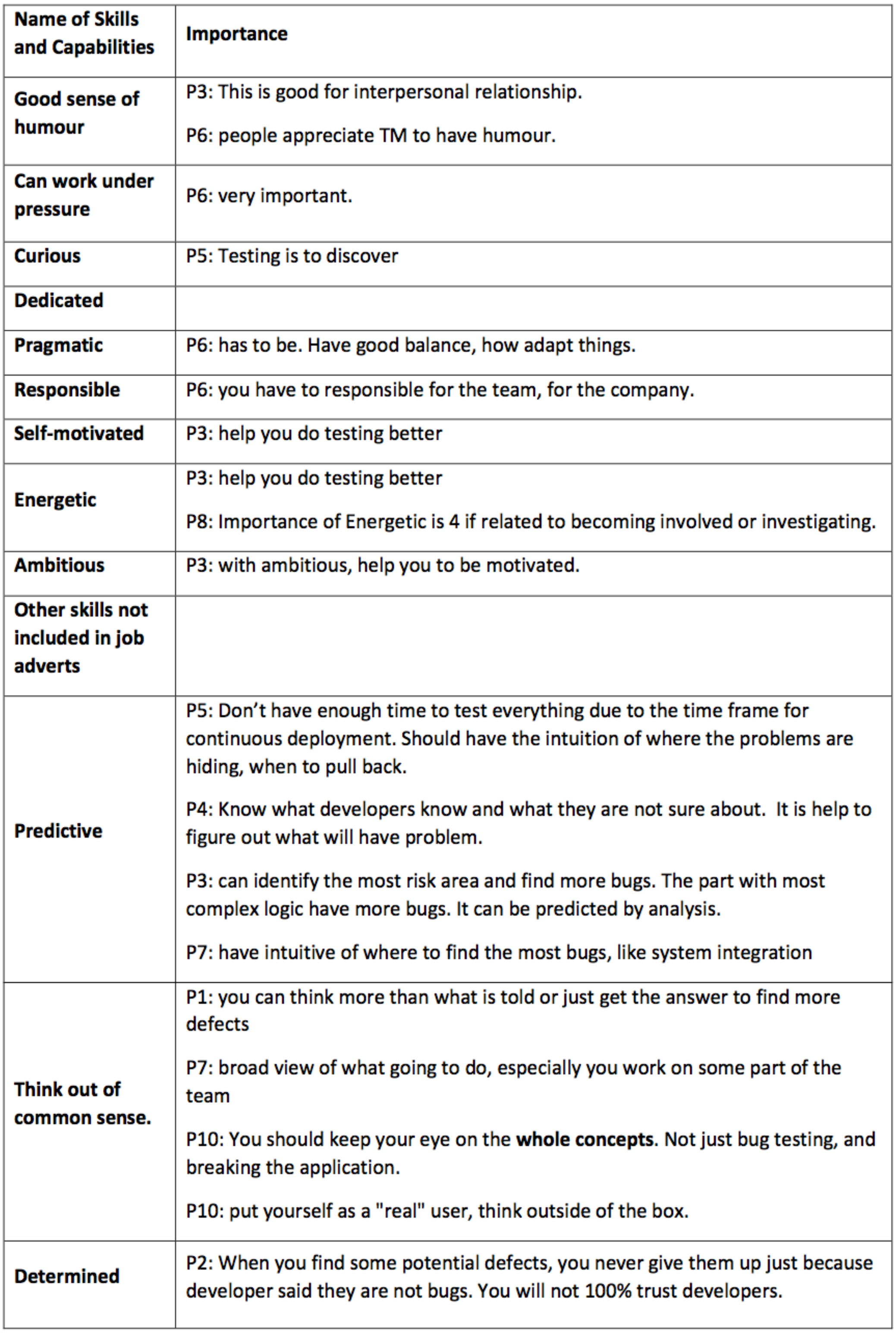
Name of Skills and Capabilities	Importance
	P6: Help to understand. Assist people in the team and make decisions. P8: Help to find the bugs. P9: If you do manual test, you can also find opportunities. But if you want to succeed in automation testing, you have to be comfortable for technical things, using different tools and different languages, CICD techniques. P10: Not just automation skills, but also knowledge in continuous deployment and Devops.
Software development process knowledge	P3: Help you to work well or smoothly with the team. P4: Can help a lot to work well in an agile team. P6 (TM): High. What process you are following, you have to think what is working or not, how to change it. P8: Because of the scrum thing, testers always talk to develops. It is important to know the workflow and process. P10: Depends on what company is doing, Waterfall or Scrum.
Database knowledge	P4: It depends. Not every company need db. P8: There are a lot of different database, you can just know some principles. P9: Depending on the work you do. P10: Depends on the applications tested
Programming language Knowledge	P4: Reading and discussing code with developers are more important than write codes. P3: This helps a lot to do automation test because they have to write code every day. P5: Having the coding knowledge helps a lot to understand the project technically and do it in a more technical way. But without the test mind-set, you can never dig into test issues. P7: Get into more programming/ understand code to see what develop do P10: Very important if doing automation test. And it is much more important to senior than intermediate.
Personal Qualities	
Passionate	P3: Without passion, you may finish the tasks, but not such good. Passion drives you to learn and make improvement.

Table G.1: Reasons for Importance Levels



Name of Skills and Capabilities	Importance
	P9: In the past, everyone can be a tester. But automation is coming in, you need to passion to know what is doing.
Adaptable	P3: You have to work with different people with different personality and culture. Although, you update your knowledge to fit the changing environment. P9: You have to be flexible with functional testing, non-functional testing,
Proactive	P9: Always
Autonomous	P1: You have to work independently P2: Intermediate and Senior all have the ability to finishes tasks independently, but senior are in the project level. P3: This is basic work ethics. P5: Basic requirements to intermediate and senior testers.
Attention to details	P3: Help you to find more defects and locate the root cause quickly P4: most important for all testers P6: not so much for TM. P9: focus on the requirements and know what to test
Positive attitude	P3: with positive attitude, you can work smoothly and corporate well with others.
Enjoys a challenge	P6: absolutely.
Customer oriented	P6: involved into clients.
Hard working	P8: Consistency is more important.
Willing to learn/share	P1: You have to keep learning new things, not only in testing, but also in software development. And never satisfied. P2: The IT techniques and methodologies updates quickly P3: techniques update fast P4: key part of tester to grow P9: You have to keep techniques new. It is always good to share the knowledge of what happens. P10: E.g. when you find a stack overflow, you cannot always find developers to solve the problems. You have to learn on you own when you need it. The reality is when you actually do it, it is a challenge and you need some experience to support.

Table G.1: Reasons for Importance Levels



Name of Skills and Capabilities	**Importance**
Good sense of humour	P3: This is good for interpersonal relationship. P6: people appreciate TM to have humour.
Can work under pressure	P6: very important.
Curious	P5: Testing is to discover
Dedicated	
Pragmatic	P6: has to be. Have good balance, how adapt things.
Responsible	P6: you have to responsible for the team, for the company.
Self-motivated	P3: help you do testing better
Energetic	P3: help you do testing better P8: Importance of Energetic is 4 if related to becoming involved or investigating.
Ambitious	P3: with ambitious, help you to be motivated.
Other skills not included in job adverts	
Predictive	P5: Don't have enough time to test everything due to the time frame for continuous deployment. Should have the intuition of where the problems are hiding, when to pull back. P4: Know what developers know and what they are not sure about. It is help to figure out what will have problem. P3: can identify the most risk area and find more bugs. The part with most complex logic have more bugs. It can be predicted by analysis. P7: have intuitive of where to find the most bugs, like system integration
Think out of common sense.	P1: you can think more than what is told or just get the answer to find more defects P7: broad view of what going to do, especially you work on some part of the team P10: You should keep your eye on the **whole concepts**. Not just bug testing, and breaking the application. P10: put yourself as a "real" user, think outside of the box.
Determined	P2: When you find some potential defects, you never give them up just because developer said they are not bugs. You will not 100% trust developers.